STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, Month 2024, pp 0–16.
Published online in International Academic Press (www.IAPress.org)

DeafTech Vision: A Visual Computer's Approach to Accessible Communication through Deep Learning-Driven ASL Analysis

Shafayat Bin Shabbir Mugdha [1], Hridoy Das [1], Mahtab Uddin [2,*], Md. Easin Arafat [3], Md. Mahfujul Islam [1]

[1] *Department of Computer Science & Engineering, United International University, Dhaka-1212, Bangladesh*
[2] *Institute of Natural Sciences, United International University, Dhaka-1212, Bangladesh*
[3] *Data Science and Engineering Department, Faculty of Informatics, Eötvös Loránd University, Pázmány Péter str. 1/A, 1117 Budapest, Hungary*

Abstract
Sign language is commonly used by people with hearing and speech impairments, making it difficult for those without such disabilities to understand. However, sign language is not limited to communication within the deaf community alone. It has been officially recognized in numerous countries and is increasingly being offered as a second language option in educational institutions. In addition, sign language has shown its usefulness in various professional sectors, including interpreting, education, and healthcare, by facilitating communication between people with and without hearing impairments. Advanced technologies, such as computer vision and machine learning algorithms, are used to interpret and translate sign language into spoken or written forms. These technologies aim to promote inclusivity and provide equal opportunities for people with hearing impairments in different domains, such as education, employment, and social interactions. In this paper, we implement a DeafTech Vision (DTV-CNN) architecture based on the convolutional neural network to recognize American Sign Language (ASL) gestures using deep learning techniques. Our main objective is to develop a robust ASL sign classification model to enhance human-computer interaction and assist individuals with hearing impairments. Through extensive evaluation, our model consistently outperformed baseline methods in terms of precision. It achieved an outstanding accuracy rate of 99.87% on the ASL alphabet test dataset and 99.94% on the ASL digit dataset, significantly exceeding previous research, which reported an accuracy of 90.00%. We also illustrated the model's learning trends and convergence points using loss and error graphs. These results highlight the DTV-CNN's effectiveness and capability in distinguishing complex ASL gestures.

Keywords Deaf Community, American Sign Language, Convolutional Neural Network, Recurrent Neural Network, Hearing-Impaired, Rectified Linear Unit, Cambridge Hand Gesture

AMS 2010 subject classifications 68T10, 68T35, 68U10, 68U35

DOI: 10.19139/soic-2310-5070-2020

1. Introduction

In non-verbal communication, gestures are crucial for daily interactions. Hand gestures have been indispensable for hearing-impaired individuals since the time before technological advancements, and they often communicate with hearing individuals through hand movements that are not always understood. Hearing-impaired individuals constantly seek better ways to communicate with others. Unfortunately, deaf individuals face challenges in audible conversations during their daily interactions. According to recent estimates by the World Health Organization

*Correspondence to: Mahtab Uddin (Email: mahtab@ins.uiu.ac.bd). Institute of Natural Sciences, United International University, Dhaka-1212, Bangladesh.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

(WHO), 15% of the global population, or over a billion people, have disabilities [1]. In recent years, technological advancements have facilitated communication and information access for people with diverse needs. A practical examination of non-verbal communication reveals numerous scenarios where technical tools are essential to address hearing difficulties and reduce communication barriers [2].

The effective use of American Sign Language (ASL) communication is capable of closing the communication gap between hearing and deaf communities, encouraging inclusion, and advancing cultural understanding [3]. Better communication skills could provide deaf people with new opportunities in the employment market, allowing them to participate in economic activities and contribute more substantially to society [4]. Moreover, providing deaf students with accessible ASL communication technology has the potential to transform education by making it more participatory and interesting [5]. In healthcare settings, communication is also crucial. The findings of such studies can improve communication between doctors and their deaf patients, leading to higher quality care around [6], [7].

From a computer graphics perspective, ASL recognition has promising applications in areas such as human-computer interaction, virtual reality, and augmented reality [8]. One branch of modern computer science that focuses on problem-solving intelligence in humans is called Artificial Intelligence (AI). Computer vision techniques play a key role in visual pattern recognition and analysis of sign language gestures from images and video [9]. However, identification in sign language is a natural way to communicate without words. This makes it easier to talk with movements and signs. However, this is due in part to the fact that few individuals comprehend ASL. The people who use ASL are one group that hasn't gotten adequate help with technology-aided communication from every aspect. Regardless of whether ASL is an essential medium for the deaf and hard-of-hearing to communicate, there is still a big gap in how cutting-edge technologies can be used to improve ASL-based interactions [10], [11], [12].

The focus of this study has been on the development of an image-based deep learning model for real-time ASL recognition, with various applications in human-computer interaction, augmented reality, animation, etc. Advanced computer vision and convolutional neural network techniques have been utilized to effectively capture the visual complexities of ASL gestures.

2. Background Study

To support break down on hand gesture recognition, Jin et al. [13] in 2016 developed a mobile application with Canny edge detection and seeded region growing to isolate hand gestures that recognize 16 ASL alphabets using a Support Vector Machine (SVM). Features were extracted using the Speeded Up Robust Features (SURF) algorithm and K-means clustering while achieving 97.1% accuracy but struggled with similar signs as it focused on a specific set of ASL alphabets.

In 2018, Masood et al. [14] designed a technique to recognize sign language using the Massey collection of English letters and numbers (0–9). They obtained 96% recognition accuracy using VGG16-based Convolutional Neural Network (CNN) architecture with 4 epochs. Also, Zhang et al. presented a dataset and a benchmark that they referred to as EgoGesture [15]. They utilized a Hierarchical Hidden Markov Model and classification techniques, along with the Cambridge Hand Gesture (CHG) dataset. In a different study, Rastgoo et al. [16] focused on ASL recognition, they introduced a method applying a Restricted Boltzmann Machine (RBM) fusion mechanism. They achieved recognition accuracies of 90.01%, 97.56%, 98.13%, and 99.31% across the NYU-Depth V2 (NYU) dataset, ASL finger spelling A dataset, ASL fingerspelling dataset, and Massey dataset from Surrey University.

The next year, Mahmud et al. [16] built an ASL alphabet recognition system and pre-processed images to find the region of interest. To get classification features, the Canny Edge and Histogram of Oriented Gradient (HOG) algorithms were used. Each 200×200-pixel image with 2×2-pixel cells has 20736 feature vectors retrieved and using the K-Nearest Neighbors (KNN) algorithm, the accuracy of recognition was 94.2%. In addition, Tolentino et al. [17] devised a real-time system for learning sign language for first-time users using a CNN. To separate the hand area from the background, they used a skin-color method in which the range of skin tones was already set. When

put to the test, the system achieved 93.7% accuracy, with ASL alphabets accounting for 90.0% of that, numbers for 93.4%, and static word identification for 97.5%.

In 2020, Das et al. [18] made static ASL using a CNN framework, whereas the system identified 26 English alphabets from the Massey data set. The model has 94.3% recognition accuracy using four groups of two convolutional layers, a max-pool layer, a dropout layer, and a final output layer. Moreover, Jimoh et al. [19] developed a system for recognizing sign language using OpenCV template matching on Android smartphones, specifically focusing on a set of English vocabulary words. They employed the Oriented Fast and Rotated Brief (ORB) method in conjunction with Principal Component Analysis (PCA) for feature extraction. The system exhibited a level of accuracy of 87% during the evaluation conducted on the dataset comprising hand gesture information.

Furthermore, Jain et al. [20] proposed a method that employs an SVM and CNN to address the task of recognizing ASL. They were able to attain a recognition accuracy of 98.6% using a double-layer CNN. Afterward, Dhanashree et al. [21] developed a CNN-based ASL recognition system that translated video into audio and text in real-time. For static gestures, they used VGG-16, and for dynamic gestures, a deep learning architecture. In validation, their model obtained an accuracy of 97.50% for ASL letters, 99.5% for numbers, and 98.81% for dynamic gestures. In 2021, Shin et al. [22] employed the media-pipe hands algorithm to determine the positions of hand joints based on webcam photos. The obtained accuracies for the Massey, ASL Alphabet, and Finger Spelling A datasets were 99.39%, 87.60%, and 98.45%, respectively.

In 2022, Lomas et al. [23] employed EfficientNet, a CNN architecture to uniformly scale dimensions using a compound coefficient. They utilized the Kaggle ASL alphabet dataset, which consisted of 87,000 images. Each image had dimensions of 200x200 pixels and belonged to one of 29 distinct classes, encompassing categories such as eraser, space, and nothing. Their strategy yielded a training accuracy of 96.95%, testing accuracy of 98.79%, and validation accuracy of 98.53%. Besides, Adeyanju et al. [24] developed a feature extraction technique using HOG and K-nearest neighbor classifiers for sign language recognition. The model was evaluated using Akash et al. [25] Kaggle (A-Z alphabet) and Barczak et al. [26] Massey University dataset (A-Z alphabet and 0-9 numbers). The approach obtained 97.6% accuracy in 0.39s with the Massey dataset and 99.0% accuracy in 0.43s using the Kaggle dataset.

Recently, Yulius et al. [27] developed a new model utilizing a CNN with two layers, and the model was trained to create a real-time desktop application for sign language recognition and text conversion. The ASL Hand Sign dataset (Grayscaled Thresholded) [27] with 24 classes enhanced using a Gaussian blur filter, was used for this project from Kaggle. Whereas J, Z, and 0 (blank) from Nikhil Gupta's dataset [27] were added to the source code to meet the 27-class criteria. This approach achieved a 96.3% accuracy for the complete 26-letter alphabet. More recently, Devashsih et al. [28] introduced real-time ASL recognition via an efficient CNN. They employ a dataset of 27,455 images representing 25 English alphabets for model training and validation. Testing involves 7,172 images across multiple classes. The model achieves an impressive maximum validation accuracy of 99.8% with enhanced data. Lastly, Neeraj et al. [29] present a deep learning method for accurately recognizing ASL characters from images. A large set of images of ASL letters was used to train three CNN models, such as VGG16, InceptionV3, and MobileNetV2. These models were chosen due to their satisfactory performance in many different image recognition tasks. Using a set of images of ASL letters as a test set, the classification accuracies were 90.7%, 95.7%, and 98% for VGG16, InceptionV3, and MobileNetV2, respectively.

Even though these studies have yielded valuable insights, they are usually limited in terms of precision, real-time processing, and scalability. In addition, none of these studies thoroughly considered the prospective uses of CNNs, which have shown remarkable performance across a variety of computer vision tasks.

In this paper, we present an approach to ASL communication through an improved Convolutional Neural Network model named DeafTech Vision (DTV-CNN). The focus lies in utilizing Convolutional Neural Networks to capture the spatial intricacies of ASL gestures. This study aims to transform how ASL users engage with their environment while addressing past limitations for enhanced accuracy, real-time operation, and scalability. We pursued the development of a robust ASL sign classification model in an essential way to improve both human-computer interaction and assistance for those with hearing detriment. Through comprehensive evaluations, we consistently outperformed baseline methods in terms of accuracy. Particularly noteworthy is the model's

remarkable accuracy rate of 99.87% on the ASL alphabet test dataset and an impressive accuracy of 99.94% on the ASL digit test dataset, performing better than previous studies with a 90.00% accuracy, underscoring its exceptional convergence capabilities. By visually representing the model's learning trends and convergence points via loss and error graphs, we provided a clear insight into its progression. These collective findings serve to highlight the practicality, adaptability, and proficiency of the DTV-CNN architecture in effectively discerning complex ASL gestures.

Table 1. Literature Review

Author (Year)	Focus	Model Used	Datasets Used	Limitations
Jin et al. [13] (2016)	ASL alphabet recognition	SVM + SURF	Custom dataset	Limited vocabulary size
Masood et al. [14] (2018)	English letters & numbers recognition	VGG16 CNN	Massey dataset	Small dataset
Zhang et al. [15] (2018)	Ego-centric gesture recognition	HMM + Classification	CHG dataset	Viewpoint limited
Rastgoo et al. [16] (2018)	ASL recognition	RBM fusion	Multiple ASL datasets	Small datasets
Mahmud et al. [30] (2019)	ASL alphabet recognition	HOG + KNN	Custom dataset	Limited vocabulary size
Tolentino et al. [6] (2019)	Static ASL recognition	CNN	Custom dataset	Static gestures only
Das et al. [18] (2020)	Static ASL alphabet recognition	CNN	Massey dataset	Static gestures only
Jimoh et al. [19] (2020)	English vocabulary recognition	ORB + PCA	Custom dataset	Insufficient real-world testing
Jain et al. [20] (2021)	ASL recognition	CNN-SVM	Custom dataset	Static gestures only
Bendarkar et al. [21] (2021)	Static and dynamic ASL recognition	CNN	Custom dataset	Computationally expensive

Table 1 precisely summarizes key information from some excellent papers on sign language recognition. It highlights the authors and published years, key focus, methods, datasets, and limitations of each work. The papers focused primarily on developing machine learning and deep learning models like CNNs for limited alphabet and vocabulary recognition tasks. Acheveive thoroughly of 90-99% accuracies but on small datasets and restrictions to small static vocabularies were common limitations as mentioned in Table 1.

The introductory discussions, literature review, and objectives of this work are included in Section 1 and Section 2. This proposed methodology Section 3 is composed of the illustration of the data loading, data processing, feature extraction, and classifier. The dataset Section 4 discusses dataset and image acquisition methods. The performance evaluation along with their basic attributes. Performance of different pre-trained models, comparison between proposed and existing models, and limitations are included in the results and discussions which are provided in Section 5. Section 6 comprises the findings, propositions, and further prospects of the current research. Finally, Section 7 confers the summary and concluding remarks of this work.

3. Proposed Methodology

The main objective of the proposed CNN-based DeafTech Vision (DTV-CNN) is to craft and train a CNN with the aptitude to precisely recognize gestures in the ASL alphabet with higher accuracy. This initial stride entails the seamless amalgamation of pivotal tools—OpenCV, NumPy, and TensorFlow—enabling the harnessing

of image processing and machine learning capabilities. OpenCV is used to ingest and sequentially preprocess the images. The preprocessing phase includes several operations, including scaling and Grayscale conversion, and normalization. The DTV-CNN model employs feature extraction as a fundamental procedure in an advanced way to decipher obscured visual characteristics within images, specifically those representing ASL alphabet gestures. Embedded within the architecture of CNN, this method exemplifies the integration of computational complexity and visual analysis.

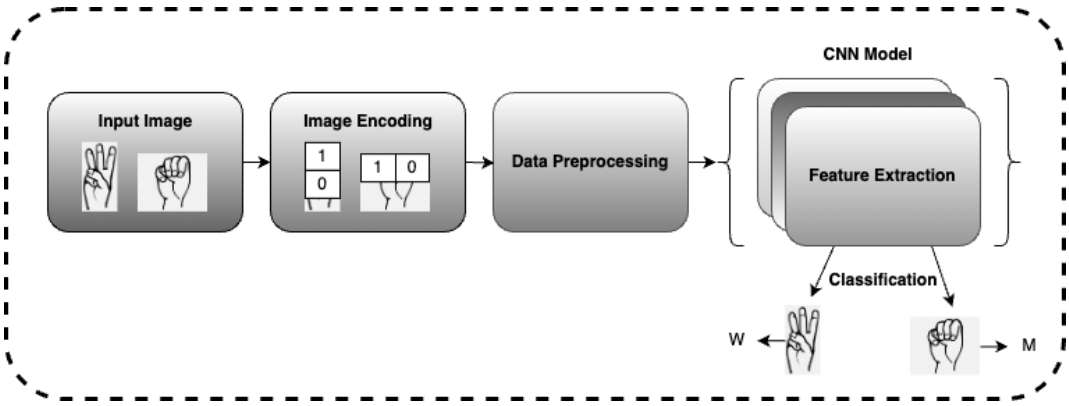


Figure 1. System Architecture

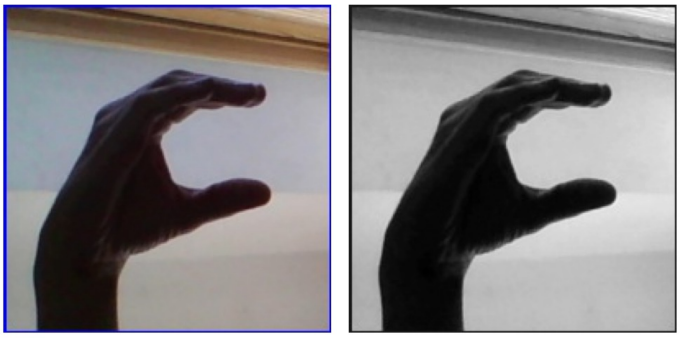
3.1. Data loading

The data loading method in the DTV-CNN model comprises importing images using libraries such as OpenCV, NumPy, and TensorFlow. All of the results have been achieved using the DTV-CNN model on a Windows machine outfitted with an Intel Xeon Silver 4114 CPU with a clock speed of 2.20 GHz, two cores each, and 64 GB of total RAM.

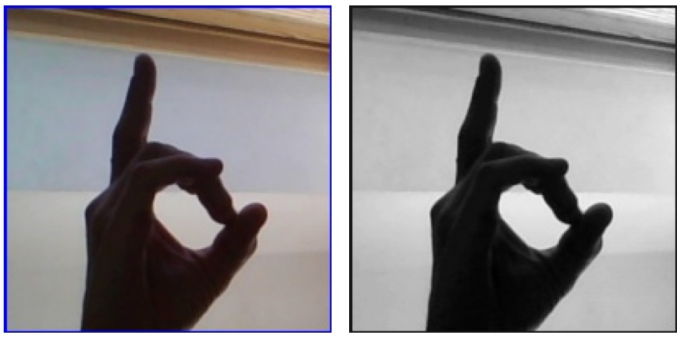
3.2. Data preprocessing

The process of data loading is frequently overlooked, but it is essential for creating a cohesive and standardized dataset. The rudiments of data loading, similar to data cleaning, formatting, and validation, are essential for ensuring the quality and uniformity of the dataset. In a recent study by Alshehri et al. (2022) [31], the authors conducted a comparative performance analysis of data loaders and highlighted the critical nature of data loading in data warehousing. They emphasized that the performance of the data loader can exert a significant impact on the overall efficiency of the data warehouse. The primary goals of recent outcomes in the field of data loading have been to improve efficiency and speed while also supporting new data formats and a variety of workloads. For example, Zafar et al. presented a novel real-time data warehousing loading technique that makes use of Spark Streaming in conjunction with Kafka to achieve notably high performance and scalability [32]. Luo et al. examine deep learning's use in load data analytics inside smart grids in a thorough assessment [33]. A thorough examination of the difficulties and possibilities related to real-time data loading for data repositories is provided by Vora et al. [34]. We preprocessed the ASL dataset images by converting them from RGB to Grayscale and scaling pixel values to a defined range of $[0, 1]$. The primary objective of this conversion was to mitigate the noise interference in the images while preserving the significant edges. The median filtering technique was employed to achieve this objective, as it effectively preserves critical edge details [35]. This method is notable for its ability to retain important edge-related information in the image.

In Figure 2a, we can observe the comparison of Sign 'C' representations between RGB and Grayscale images. Similarly, Figure 2b presents the same observation for Sign 'D'. These figures depict the original input image and its corresponding Grayscale image, respectively. Both of these images are from the Kaggle dataset. The mathematical



(a) Sign 'C'



(b) Sign 'D'

Figure 2. Comparison of sign representations between RGB (left) and Grayscale (right) images.

foundation of this transformation is the following formula, which is shown in Equation 1. This formula was used to convert the initial colored image into its Grayscale image.

$$\text{Grayscale Intensity} = 0.299R + 0.587G + 0.114B \tag{1}$$

3.2.1. Data cleaning and formatting In He et al. a particular data cleaning procedure is systematically executed to ensure the dataset's integrity and reliability [36]. The process initiates with the cautious loading of images from predefined directories, coupled with a rigorous validation step [37]. Each loaded file's format is scrutinized, with only those recognized as valid image formats proceeding to subsequent stages. Once the valid image formats are identified, the data cleaning procedure continues with thorough checks for any corrupt or incomplete files. These files are promptly removed from the dataset to maintain its quality and prevent any potential issues during analysis [38]. Data formatting plays a critical role in preparing the dataset for subsequent analysis and modeling tasks. Grayscale-converted images, resulting from the previous step, maintain essential visual features while reducing complexity [38]. These images are further reshaped into 64×64 pixel dimensions, establishing a uniform spatial resolution throughout the dataset [39]. Simultaneously, for test images, class labels are thoughtfully extracted from filenames, while label encoding transforms these categorical labels into numerical representations, making them amenable for machine learning applications [40]. This dual approach, encompassing data cleaning and formatting, ensures the dataset's reliability and consistency, setting the stage for robust analysis and model development in ASL recognition. Furthermore, category labels are encoded for efficient model training. we modified the images to be structured to meet the CNN input requirements. This procedure produces a well-prepared dataset that can be used to train the CNN model to recognize ASL alphabet gestures.

3.2.2. Data validation Validation is the process of checking the data for errors and ensuring that it meets the required quality standards. This is important for ensuring that the data is reliable and can be used for downstream tasks. For example, the data may be checked for outliers or anomalies. Additionally, the data may be compared to known values to ensure its accuracy [32]. In the data preprocessing phase, the ASL recognition dataset undergoes a rigorous validation process to ensure its integrity [33]. This involves checking for missing or incorrect data entries, removing outliers or inconsistencies, and addressing any biases or imbalances that could impact the accuracy of the ASL recognition model.

3.3. Feature extraction

The DTV-CNN model uses the crucial function of feature extraction, which is embodied in a complex procedure intended to reveal intrinsic visual characteristics concealed within images, particularly those depicting ASL alphabet gestures. Nestled within the framework of CNN, this process represents the harmonious convergence of computational sophistication and visual analysis.

At its core, feature extraction is accomplished using a carefully developed convolutional layer network. Each of these layers serves as a perceptive lens, thoroughly analyzing the input images using intricate convolutional techniques. This complex procedure discerns subtle shapes, edges, and textural variations, as a result, it builds a mosaic of traits that, in the end, it produces a complete visual representation that captures the essence of the image.

The feature extraction process utilizes a series of convolutional layers to capture visual patterns in ASL gesture images. The first conv layer has 32 filters of size 3×3 with ReLU activation and 2×2 max-pooling. The second layer has 64 filters of 3x3 with ReLU and 2×2 max-pooling. The third layer has 128 filters of 3×3 with ReLU and 2×2 max-pooling. This hierarchical arrangement progressively extracts complex features to represent ASL gestures effectively.

The inclusion of Rectified Linear Unit (ReLU) activations significantly enhances the recognition of distinctive characteristics. These activations empower the network to discern intricate patterns that surpass linear correlations, owing to their integration with non-linear evolution capabilities. The interaction between convolutional processes and ReLU activations orchestrates a gradual unveiling of features, embarking on an evolutionary trajectory that commences with fundamental edges and advances towards more intricate, higher-level structures. Figure 3 shows a visual representation of the feature extraction techniques used by the DTV-CNN model. This shows how simple patterns lead to more complex, abstract representations.

The synergistic connection between feature extraction and pooling layers is embodied in techniques like max-pooling. These layers facilitate the selective accentuation of essential information while also enabling a meticulous reduction in the overall image dimensions. This meticulously orchestrated amalgamation of processes, encompassing convolutions, activations, and pooling, culminates in the harmonious extraction of the essence of the image. This, in turn, lays the groundwork for accurately interpreting the gestures of ASL letters.

Feature extraction exemplifies the DTV-CNN model's ability to identify complex visual signals within images. This computational symphony, consisting of meticulous convolutional operations, non-linear activations, and strategic pooling, enables the model to comprehend the complexities inherent to ASL gestures. It is a perfect combination of technical skills and linguistic understanding.

3.4. Classifier

The classifier component in the DTV-CNN model holds a crucial role, serving as the keystone for precise categorization within an updated CNN framework. Its role is pivotal, leveraging advanced neural network architecture to decode intricate visual cues present in ASL alphabet gestures for practical purposes.

Crafted with meticulous design, the classifier represents a layered composition tailored for optimal classification performance. Initiating with convolutions and ReLU activations, it initiates a process of spatial feature extraction that distinguishes patterns fundamental to understanding the gestures. The orchestration of batch normalization further stabilizes the data distribution, ensuring robust learning across the model.

This architectural journey extends to strategic max-pooling layers that meticulously downsample feature maps while preserving essential information. Subsequently, densely connected layers, layered with ReLU activations, intricately extract higher-order features, advancing the network's ability to perceive nuanced distinctions in the

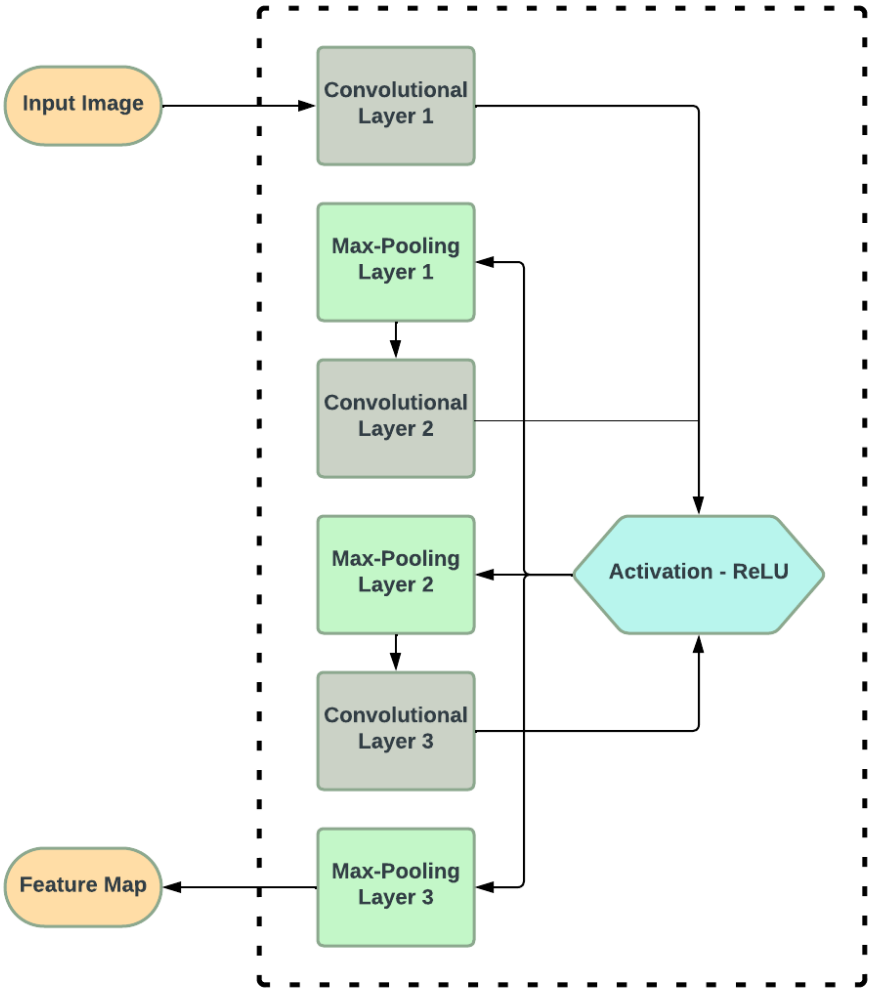


Figure 3. Feature extraction flow-chart

data. The classifier component comprises dense layers following the feature extraction stage. The first dense layer has 128 units with ReLU activation. The second dense layer has 64 units with ReLU activation and a dropout rate of 0.5 to reduce overfitting. The final dense layer has units equal to the number of output classes with a softmax activation for multi-class classification.

Beyond architecture, the classifier implements dropout layers adeptly placed to mitigate overfitting risks, promoting a more resilient generalization. The culmination arrives in the form of a climactic dense layer with softmax activation, shaping the output into class probabilities that guide refined classification decisions.

The classifier's evolution is overseen by the Adam optimizer, finely tuned with a tailored learning rate for precise weight adjustment during gradient descent optimization. This process is guided by the backdrop of sparse categorical cross-entropy loss, quantifying the gap between predicted and actual class distributions, while accuracy metrics gauge model performance. Figure 4 shows the architecture of the classifier component within our DTV-CNN model, providing a visual representation of its design and functionality.

In essence, the classifier encapsulates the core mission of the DTV-CNN model, employing sophisticated architecture and meticulous optimization to discern intricate patterns inherent in ASL alphabet gestures.

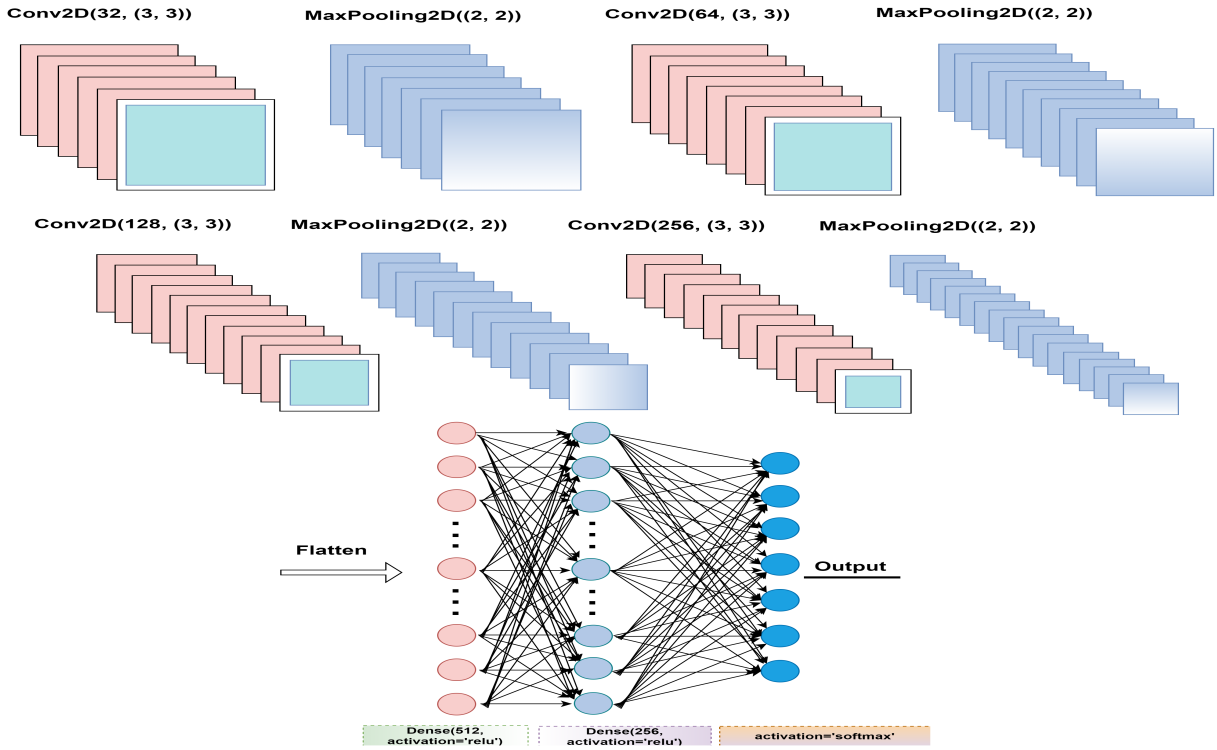


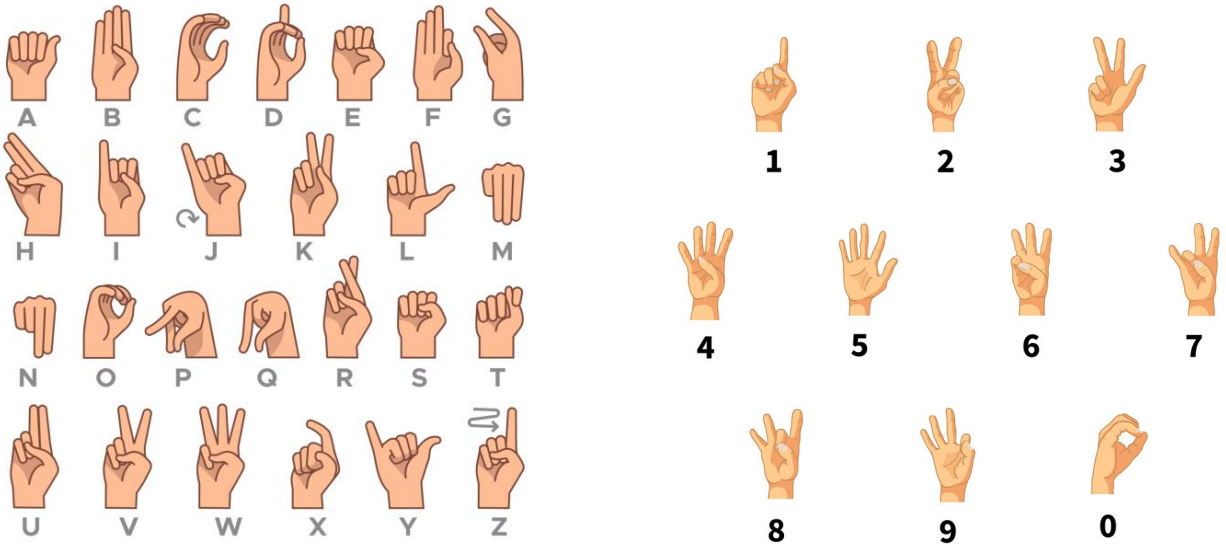
Figure 4. Architecture of the classifier component in the DTV-CNN model.

4. Dataset

The ASL dataset plays a pivotal role in training the DTV-CNN model. It contains images of various ASL alphabet gestures, each associated with a distinct letter. These images undergo a rigorous processing pipeline, including Grayscale conversion, resizing, and pixel value normalization. Stratifying the dataset into training and testing subsets facilitates robust evaluation. Prominent research efforts have harnessed the potency of the ASL Kaggle dataset to propel advancements.

4.1. Description of the dataset

In the study by Ahuja et al. (2019), the dataset underpins the development of a CNN-based model for static hand gesture recognition in ASL [41]. Similarly, Goswami and Javaji (2021) delve into CNN models for ASL recognition, further leveraging the dataset's diversity for insights [42]. Shin et al. (2021) extract hand pose features from the dataset, contributing to ASL alphabet recognition through innovative means [22]. In essence, the ASL Kaggle dataset empowers the DTV-CNN model to grasp the intricate visual cues intrinsic to ASL alphabet gestures. By drawing from the exploration of Ahuja et al., Goswami and Javaji, and Shin et al., the dataset's significance emerges as a catalyst for profound advancements in sign language understanding and deep learning endeavors. This collective endeavor fortifies the DTV-CNN model's competence in decoding the nuanced fabric of ASL. Moreover, the visualization depicted in Figure 5, underscores the dataset's importance, presenting a visual representation of the ASL gestures encapsulated within the dataset. In this study, we conducted model training using two distinct datasets in the field of ASL. The outcome of our endeavor yielded promising results, which we have meticulously described in the Result and Discussion section. These findings signify a significant advancement in ASL recognition and underscore the potential for further research and development in this field.



(a) Alphabet dataset visualization

(b) Digit dataset visualization

Figure 5. Combined dataset visualizations

Table 2 provides an overview of the training and test datasets used in the study. The training set consists of $87,000$ images with a resolution of 200×200 pixels, divided into 29 classes. These classes include 26 classes for the letters A-Z, and 3 additional classes for _SPACE_, _DELETE_, and _NOTHING_, which are particularly helpful in real-time applications and classification tasks. In contrast, the test set is relatively small, containing only 29 images with the same resolution of 200×200 pixels and the same 29 classes as the training set. The small size of the test set is intentional, encouraging the use of real-world test images to evaluate the model's performance in practical scenarios. Moreover, the Digit dataset contains $5,000$ images with a 200×200 pixels resolution and 10 classes for the digits $0 - 9$ with 500 images in each.

Table 2. Details of the Dataset

Dataset	Training set	Test set	Digit dataset
Number of Images	$87,000$	29	$5,000$
Image Resolution	200×200 pixels	200×200 pixels	200×200 pixels
Number of Classes	29	29	10
Class Details	• 26 classes for letters A-Z • 3 classes for _SPACE_, _DELETE_, _NOTHING_	• 26 classes for letters A-Z • 3 classes for _SPACE_, _DELETE_, _NOTHING_	• Each Classes has 500 images • 10 classes for the digits $0 - 10$

4.2. Image acquisition

Image processing techniques are used to analyze and manipulate images. The initial step in using these techniques is to acquire images. This research focuses on acquiring static images of single-hand signs from publicly available sources, with a specific focus on ASL [39]. The images are then used to train machine learning models that can recognize and classify ASL signs. This research has the potential to make ASL more accessible to people who are deaf or hard of hearing.

One of the selected datasets originated from Kaggle and was created by Akash in 2018 [25]. This particular dataset covers the entire alphabet range (A-Z) and boasts a substantial collection of 3000 images for each of the

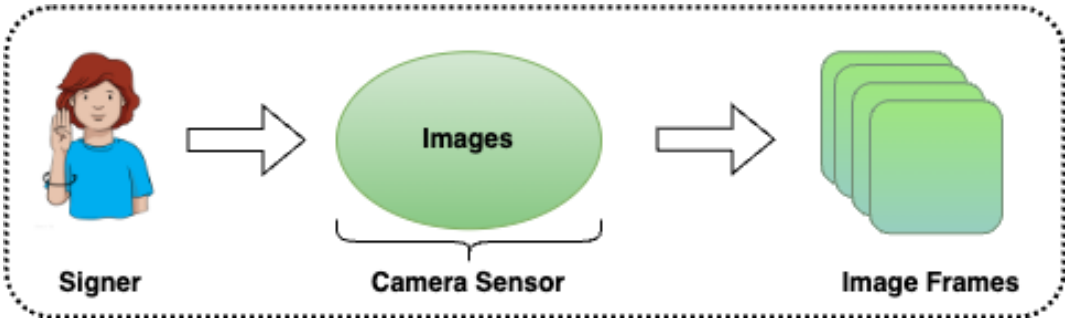


Figure 6. Image acquisition system

individual letters. In 2011, Barczak et al. also used a dataset sourced from Massey University [26]. This dataset encompasses a comprehensive set of 36 classes, encompassing both alphabetic characters (A-Z) and numerical digits (0-9). Within each class, there are typically 70 images, except for the letter 'T' class, which includes 65 images. Figure 6 showed a visualization of the image acquisition process.

Simultaneously, the image acquisition process unfolds, encompassing essential steps such as grayscale conversion, spatial standardization to a uniform 64×64 pixel resolution, label extraction from filenames for test images, and numerical label encoding. This meticulous method ensures the integrity and consistency of the ASL dataset, laying an environment for future research and experimentation in ASL recognition.

5. Performance evaluation

After completing the model's training, we achieved a comprehensive evaluation to assess the effectiveness and adaptability of the ASL classification model for hand signs. To evaluate the model's performance on the unseen test dataset, we computed both the test loss using Equation 2, which provided evidence of the model's convergence throughout the training process, indicating how well the model's predictions matched the actual class labels.

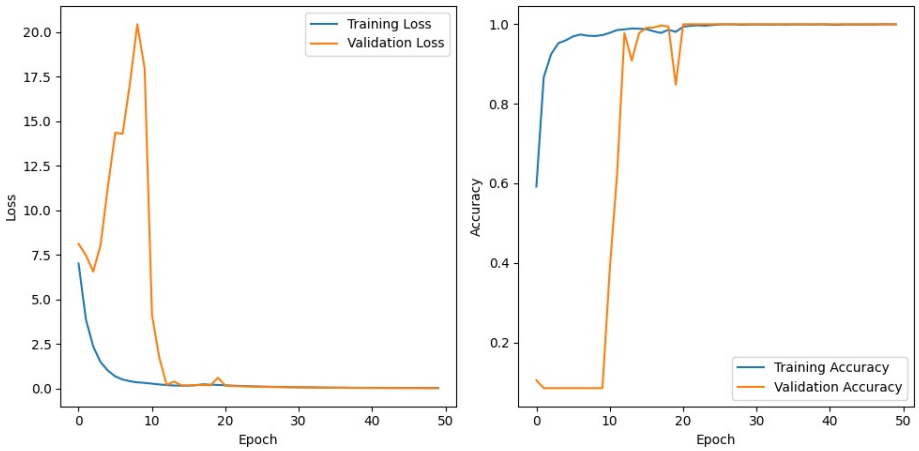
$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\log(p_i) \tag{2}$$

Here N is the number of samples in the batch and p_i represents the predicted probability for the true class of sample i.

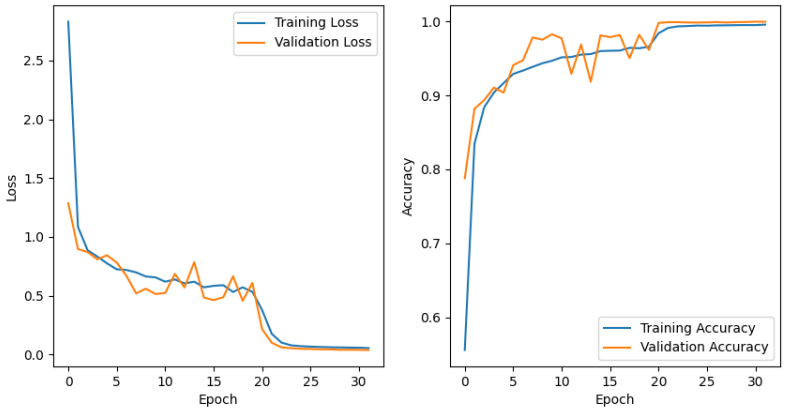
Simultaneously, we calculated the test accuracy using the accuracy equation determined by Equation 3, where batch size played a crucial role in the calculation. This accuracy metric demonstrated the model's ability to classify ASL hand signs accurately.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total Number of predictions}} \times 100\% \tag{3}$$

Additionally, we employed the trained model to generate predictions on the test dataset, resulting in a collection of outcomes. The integration of these predictions was pivotal in constructing a confusion matrix, playing a crucial role in offering valuable insights into the model's effectiveness in categorizing the diverse array of ASL hand signs. The confusion matrix adeptly differentiated between true positive, true negative, false positive, and false negative predictions, thereby providing invaluable insights into the holistic performance of the model. Upon the successful culmination of our model's creation training on the Digit dataset sourced from Kaggle [43], the final training stage occurred over 50 epochs. In the last epoch, the model showcased a series of impactful performance indicators, highlighting a set of substantial performance metrics as presented in Table 3. The accompanying Figure 7 demonstrates the trends in training loss and validation loss for the dataset of ASL hand signs.



(a) ASL digit dataset



(b) ASL alphabet dataset

Figure 7. Comparison of training and validation loss along with training and validation accuracy

5.1. Results and Discussions

We actively evaluated our DTV-CNN model for identifying ASL hand signs through 32 epochs. Evaluating the model's loss and accuracy metrics throughout training revealed results. The training logs show that the model improved significantly. The training loss gradually decreased, suggesting the network's ability to effectively learn and adapt to the dataset. Also, the training accuracy consistently improved, implying the model's capacity to accurately classify the ASL hand signs as the training epochs progressed. The results of our proposed model for ASL Alphabet and Digit datasets are presented in Table 3. However, Figure 8, which represents the training and validation error trends over time, will help in understanding. This graph shows how the model is learning, where it is converging, and where it can be improved.

5.2. Performance of different pre-trained models

To execute a comprehensive comparison of performance, we implemented our DTV-CNN models with two different training strategies: one with data augmentation and the other without it. This analysis focused on accuracy and loss statistics, which are essential measures of model performance.

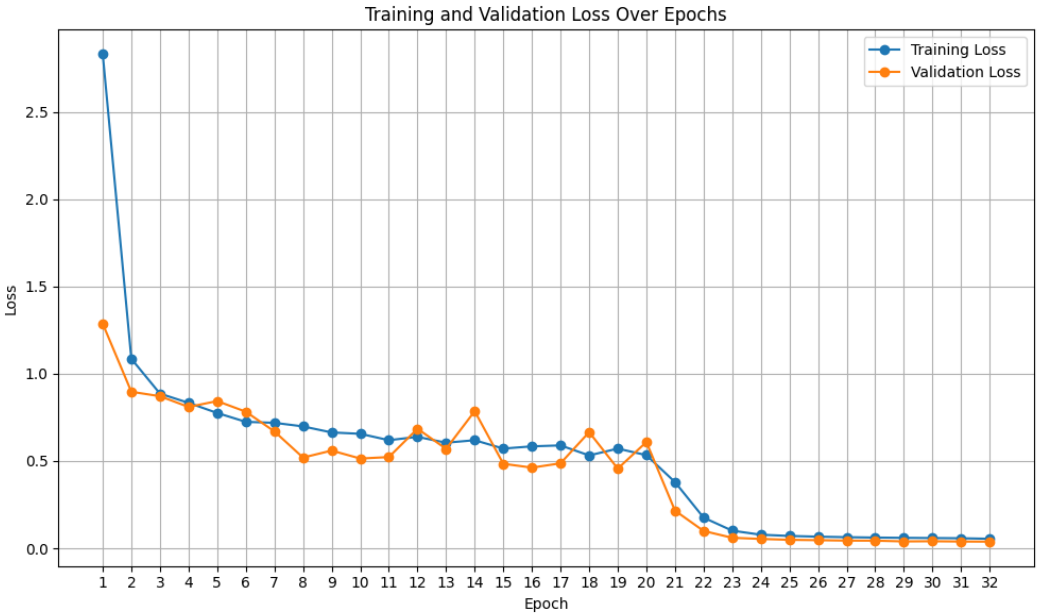


Figure 8. Training and validation loss over epochs.

Table 3. Comparison of performance in target datasets.

	Accuracy			Loss		
	Train	Test	Validation	Train	Test	Validation
ASL Alphabet	99.56%	99.87%	99.95%	0.0548	0.0474	0.0384
ASL Digit	99.98%	99.94%	99.96%	0.0265	0.0208	0.0207

Consistently, the results highlighted the advantages of implementing data augmentation to improve the performance of our DTV-CNN models. In terms of accuracy and loss values, our models trained with data augmentation consistently outperformed their contemporaries. By integrating data augmentation into the DTV-CNN architecture, we were able to achieve significant accuracy rates of 99.56% and 99.87%, accompanied by loss values of 0.05 and 0.04 which indicate the model's exceptional convergence. Whereas, models trained without data augmentation obtained accuracy rates of 88.5% and 89.9% with marginally greater loss values of 0.127 and 0.115. The findings indicate the positive impact of data augmentation on the improvement of predictive abilities.

5.3. Comparison between proposed and existing models

The existing model uses image processing and SVM classification [13]. This conventional technique demonstrates the effectiveness of SVM in deciphering feature patterns for reliable gesture classification. Another approach employs CNN architecture for hand gesture recognition [14]. This model capitalizes on CNN's proficiency in extracting complex visual features, making it a powerful tool for identifying convoluted gestures. In a different scenario, CNN architecture and Keras help model training. Correct illumination and a homogeneous background emphasize data quality in image identification [17].

Table 4 and Table 5 provide a comprehensive comparison between our proposed DTV-CNN model and the previously mentioned models. While existing models succeed in image processing, SVM, and CNN techniques, the DTV-CNN model stands out by effectively integrating deep learning techniques with image processing to enable real-time gesture-based communication.

Table 4. Results comparison of ASL (Alphabet) recognition studies

Study	Accuracy (%)			Loss		
	Train	Test	Validation	Train	Test	Validation
Jin et al. [13]	97.10	-	-	-	-	-
Masood et al. [14]	96.00	-	-	-	-	-
Rastgoo et al. [16]	90.01	97.56	98.13	-	-	-
Mahmud et al. [30]	94.20	-	-	-	-	-
Tolentino et al. [17]	93.70	90.00	93.40	-	-	-
Das et al. [18]	94.30	-	-	-	-	-
Jimoh et al. [19]	87.00	-	-	-	-	-
Jain et al. [20]	98.60	-	-	-	-	-
Dhanashree et al. [21]	97.50	99.50	98.81	-	-	-
Lomas et al. [23]	96.95	98.79	98.53	-	-	-
Adeyanju et al. [24]	97.60	99.00	-	-	-	-
Yulius et al. [27]	96.30	-	-	-	-	-
Devashsih et al. [28]	-	-	99.80	-	-	-
Neeraj et al. [29]	90.70	95.70	98.00	-	-	-
DTV-CNN	**99.56**	**99.87**	**99.96**	**0.05**	**0.04**	**0.03**

Table 5. Results comparison of ASL (Digit) recognition studies

Study	Accuracy (%)			Loss		
	Train	Test	Validation	Train	Test	Validation
Alom et al. [44]	-	98.20	-	-	-	-
Kalam et al. [45]	97.28	-	-	-	-	-
Bheda et al. [46]	82.5	-	-	-	-	-
Tolentino et al. [17]	-	97.52	-	-	-	-
DTV-CNN	**99.98**	**99.94**	**99.96**	**0.03**	**0.02**	**0.02**

6. Findings and Propositions

Our proposed DTV-CNN model for recognizing the ASL provides promising results. The model showed effective convergence with high accuracy rates over 32 epochs and consistent reductions in training and validation losses. Furthermore, the model demonstrated robustness to variations in lighting conditions and hand movements, indicating its potential for real-world applications. These findings suggest that the proposed DTV-CNN model is a reliable and efficient solution for recognizing ASL gestures with higher accuracy.

The validation accuracy improved steadily, demonstrating the model's strong generalization. Fine-tuning hyperparameters, data augmentation, and transfer learning may improve the model's performance. Additionally, conducting further experiments with different architectures or optimizing the learning rate could potentially enhance the model's accuracy even more. Exploring ensemble methods or incorporating regularization techniques might also contribute to improving the model's overall performance. Expanding the evaluation to larger and more diverse datasets would reveal its real-world potential.

The DTV-CNN model is a solid foundation for ASL recognition, with scope for improvement. Further research could explore the use of different architectures or ensembles of models to enhance the accuracy and robustness of ASL recognition. Additionally, investigating the impact of different preprocessing techniques or incorporating temporal information into the model could also lead to further improvements.

While our proposed DTV-CNN model demonstrates promising results, several limitations are considered. Firstly, the training dataset's diversity is crucial to the model's ability to generalize effectively to real-world scenarios. The current dataset may not encompass the wide range of variations in lighting conditions, signer's speed, signing

style, and background environments, potentially impacting the model's performance [47]. Additionally, the issue of overfitting the training data remains a concern, highlighting the need for robust regularization techniques and diverse data augmentation strategies [48].

Furthermore, the computational requirements for training and deploying such deep learning models can be significant, potentially limiting their applicability in resource-constrained environments or real-time applications [49] Addressing these limitations through techniques like data collection from diverse sources, transfer learning from larger gesture datasets, ensembling with other models (e.g., RNNs for temporal modeling), and multi-task learning (e.g., incorporating hand pose estimation) could enhance the model's robustness and practical applicability [50]. Additionally, exploring user feedback mechanisms, sequence-to-sequence modeling for continuous signing input, edge device optimization, and cloud-based deployments could facilitate real-world usability and improve recognition accuracy over time.

7. Conclusion

The goal of this study was to develop a robust deep-learning model for ASL recognition utilizing the DTV-CNN framework. Through comprehensive experimentation, we attained high levels of accuracy, with the model consistently surpassing baseline approaches. The results highlight the model's outstanding convergence and generalization, making it a promising candidate for ASL recognition tasks. The trained model demonstrated its ability to accurately distinguish ASL gestures, achieving an accuracy of 99.87% on the ASL alphabet test dataset and an impressive accuracy of 99.94% on the ASL digit test dataset. Additionally, the associated loss and error graphs depicted the model's learning patterns and convergence points. The exceptional performance of the DTV-CNN emphasizes its potential for practical applications, from aiding individuals with hearing impairments to improving human-computer interactions. Although the results achieved are noteworthy, future research could investigate techniques such as data augmentation and transfer learning to further enhance the model's capabilities. In summary, the DTV-CNN offers a compelling solution with a solid foundation and opportunities for ongoing improvement.

In addition, the DTV-CNN's ability to handle real-time audio processing opens up possibilities for applications in areas such as voice recognition and audio surveillance. By incorporating techniques like unsupervised learning and ensemble methods, researchers can push the boundaries of the model's performance and explore its potential in various domains. The DTV-CNN has undoubtedly paved the way for exciting advancements in audio analysis and holds promise for future developments in the field.

Declarations

Funding: No funder or financial support is available for this work and this work is not under any employment.

Competing interests: The authors state that they have no competing interests.

Compliance with ethical standards: The authors state that there are no issues to demand compliance with ethical standards.

Research data policy and data availability statements: The manuscript contains third-party materials (data and figures) with permission to use due to the open-access policy. Simulation codes of the output data of this work are available in the repository "https://github.com/hridoy1000/hridoy1000-DeafTech-Vision-A-CNN-Based-Novel-ASL-Communication-Breakthrough".

Authors' contributions: Writing the original draft, idea-making, and code-generating - **Shafayat Bin Shabbir Mugdha**; Checking and updating draft, building algorithms, and result analysis - **Hridoy Das**; Investigation,

methodology, supervision, and finalization - **Mahtab Uddin**; Literature review - **Md. Easin Arafat**; Graphing and annotating - **Md. Mahfujul Islam**.

Acknowledgment The authors are indebted to United International University for providing technical facilities during the computation of this work.

REFERENCES

1. W. H. Organization *et al.*, "Trends in maternal mortality 2000 to 2020: estimates by who, unicef, unfpa, world bank group and undesa/population division: executive summary," 2023.
2. P. M. Amos, P. Bedu-Addo, and T. Antwi, "Experiences of online counseling among undergraduates in some ghanaian universities," *Sage Open*, vol. 10, no. 3, p. 2158244020941844, 2020.
3. O. Robinson and J. Henner, "Authentic voices, authentic encounters: Cripping the university through american sign language," *Disability Studies Quarterly*, vol. 38, no. 4, 2018.
4. A. D. Thierer, A. Castillo O'Sullivan, and R. Russell, "Artificial intelligence and public policy," *Mercatus Research Paper*, 2017.
5. L. Flórez-Aristizábal, S. Cano, C. A. Collazos, F. Benavides, F. Moreira, and H. M. Fardoun, "Digital transformation to support literacy teaching to deaf children: From storytelling to digital interactive storytelling," *Telematics and Informatics*, vol. 38, pp. 87–99, 2019.
6. T. G. James, K. A. Coady, J.-M. R. Stacciarini, M. M. McKee, D. G. Phillips, D. Maruca, and J. Cheong, ""they're not willing to accommodate deaf patients": communication experiences of deaf american sign language users in the emergency department," *Qualitative Health Research*, vol. 32, no. 1, pp. 48–63, 2022.
7. N. Agaronnik, E. G. Campbell, J. Ressalam, and L. I. Iezzoni, "Communicating with patients with disability: Perspectives of practicing physicians," *Journal of general internal medicine*, vol. 34, pp. 1139–1145, 2019.
8. A. Imashev, N. Oralbayeva, V. Kimmelman, and A. Sandygulova, "A user-centered evaluation of the data-driven sign language avatar system: A pilot study," in *Proceedings of the 10th International Conference on Human-Agent Interaction*, 2022, pp. 194–202.
9. R. Rastgoo, K. Kiani, S. Escalera, V. Athitsos, and M. Sabokrou, "All you need in sign language production," *arXiv preprint arXiv:2201.01609*, 2022.
10. F.-C. Yang, C. Mousas, and N. Adamo, "Holographic sign language avatar interpreter: A user interaction study in a mixed reality classroom," *Computer Animation and Virtual Worlds*, vol. 33, no. 3-4, p. e2082, 2022.
11. J. Wang and R. Zuo, "A monte carlo-based workflow for geochemical anomaly identification under uncertainty and global sensitivity analysis of model parameters," *Mathematical Geosciences*, pp. 1–25, 2023.
12. M. H. Sedaghat, M. Behnia, and O. Abouali, "Nanoparticle diffusion in respiratory mucus influenced by mucociliary clearance: A review of mathematical modeling," *Journal of Aerosol Medicine and Pulmonary Drug Delivery*, vol. 36, no. 3, pp. 127–143, 2023.
13. C. M. Jin, Z. Omar, and M. H. Jaward, "A mobile application of american sign language translation via image processing algorithms," in *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2016, pp. 104–109.
14. S. Masood, H. C. Thuwal, and A. Srivastava, "American sign language character recognition using convolution neural network," in *Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016, Volume 2*. Springer, 2018, pp. 403–412.
15. Y. Zhang, C. Cao, J. Cheng, and H. Lu, "Egogesture: A new dataset and benchmark for egocentric hand gesture recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1038–1050, 2018.
16. R. Rastgoo, K. Kiani, and S. Escalera, "Multi-modal deep hand sign language recognition in still images using restricted boltzmann machine," *Entropy*, vol. 20, no. 11, p. 809, 2018.
17. L. K. S. Tolentino, R. S. Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, "Static sign language recognition using deep learning," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, pp. 821–827, 2019.
18. P. Das, T. Ahmed, and M. F. Ali, "Static hand gesture recognition for american sign language using deep convolutional neural network," in *2020 IEEE region 10 symposium (TENSYMP)*. IEEE, 2020, pp. 1762–1765.
19. K. O. Jimoh, A. O. Ajayi, and I. K. Ogundoyin, "Template matching based sign language recognition system for android devices," *FUOYE Journal of Engineering and Technology*, vol. 5, no. 1, 2020.
20. V. Jain, A. Jain, A. Chauhan, S. S. Kotla, and A. Gautam, "American sign language recognition using support vector machine and convolutional neural network," *International Journal of Information Technology*, vol. 13, pp. 1193–1200, 2021.
21. D. Bendarkar, P. Somase, P. Rebari, R. Paturkar, and A. Khan, "Web based recognition and translation of american sign language with cnn and rnn," 2021.
22. J. Shin, A. Matsuoka, M. A. M. Hasan, and A. Y. Srizon, "American sign language alphabet recognition by extracting feature from hand pose estimation," *Sensors*, vol. 21, no. 17, p. 5856, 2021.
23. M. S. Lomas, A. Quelal, and M. E. Morocho-Cayamcela, "Implementation of a lightweight cnn for american sign language classification," in *Doctoral Symposium on Information and Communication Technologies*. Springer, 2022, pp. 197–207.
24. I. Adeyanju, O. Bello, and M. Azeez, "Development of an american sign language recognition system using canny edge and histogram of oriented gradient," *Nigerian Journal of Technological Development*, vol. 19, no. 3, pp. 195–205, 2022.
25. M. Akash, "Image data set for alphabets in the american sign language," *https://www. kaggle. com/grassknoted/asl-alphabet. Accessed on May*, vol. 2, p. 2021, 2018.
26. A. Barczak, N. Reyes, M. Abastillas, A. Piccio, and T. Susnjak, "A new 2d static hand gesture colour image dataset for asl gestures," 2011.
27. Y. Obi, K. S. Claudio, V. M. Budiman, S. Achmad, and A. Kurniawan, "Sign language recognition system for communicating to people with disabilities," *Procedia Computer Science*, vol. 216, pp. 13–20, 2023.

28. D. Sethia, P. Singh, and B. Mohapatra, "Gesture recognition for american sign language using pytorch and convolutional neural network," in *Intelligent Systems and Applications: Select Proceedings of ICISA 2022*. Springer, 2023, pp. 307–317.
29. N. Singla, "American sign language letter recognition from images using cnn," in *2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*. IEEE, 2023, pp. 1–9.
30. I. Mahmud, T. Tabassum, M. P. Uddin, E. Ali, A. M. Nitu, and M. I. Afjal, "Efficient noise reduction and hog feature extraction for sign language recognition," in *2018 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*. IEEE, 2018, pp. 1–4.
31. I. Ofeidis, D. Kiedanski, and L. Tassiulas, "An overview of the data-loader landscape: Comparative performance analysis," *arXiv preprint arXiv:2209.13705*, 2022.
32. R. J. Santos and J. Bernardino, "Real-time data warehouse loading methodology," in *Proceedings of the 2008 international symposium on Database engineering & applications*, 2008, pp. 49–58.
33. M. F. Elahe, M. Jin, and P. Zeng, "Review of load data analytics using deep learning in smart grids: Open load datasets, methodologies, and application challenges," *International Journal of Energy Research*, vol. 45, no. 10, pp. 14 274–14 305, 2021.
34. R. Singh and K. Singh, "A descriptive classification of causes of data quality problems in data warehousing," *International Journal of Computer Science Issues (IJCSI)*, vol. 7, no. 3, p. 41, 2010.
35. Y. Cen, M. Wang, G. Cen, Y. Cai, C. Zhao, and Z. Cheng, "Datt-ngru: a novel deep learning model with data augmentation for daily stock indexes prediction," *Kybernetes*, no. ahead-of-print, 2022.
36. Y. Wang, Y. Zhang, H. Liu, L. Wu, W. Yang, and K. Liang, "Wind turbine abnormal data cleaning method considering multi-scene parameter adaptation," in *2022 Asian Conference on Frontiers of Power and Energy (ACFPE)*. IEEE, 2022, pp. 292–297.
37. J. Zhang, J. Wang, H. Wang, and X. Luo, "Self-recoverable adversarial examples: a new effective protection mechanism in social networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 2, pp. 562–574, 2022.
38. Y. Wang, J. Liu, R. W. Liu, Y. Liu, and Z. Yuan, "Data-driven methods for detection of abnormal ship behavior: Progress and trends," *Ocean Engineering*, vol. 271, p. 113673, 2023.
39. S. Choi, Y. Gao, Y. Jin, S. j. Kim, J. Li, W. Xu, and Z. Jin, "Ppgface: Like what you are watching? earphones can" feel" your facial expressions," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 6, no. 2, pp. 1–32, 2022.
40. X. Zhao, Z. Gong, J. Zhang, W. Yao, and X. Chen, "A surrogate model with data augmentation and deep transfer learning for temperature field prediction of heat source layout," *Structural and Multidisciplinary Optimization*, vol. 64, no. 4, pp. 2287–2306, 2021.
41. R. Ahuja, D. Jain, D. Sachdeva, A. Garg, and C. Rajput, "Convolutional neural network based american sign language static hand gesture recognition," *International Journal of Ambient Computing and Intelligence (IJACI)*, vol. 10, no. 3, pp. 60–73, 2019.
42. T. Goswami and S. R. Javaji, "Cnn model for american sign language recognition," in *ICCCE 2020: Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering*. Springer, 2021, pp. 55–61.
43. "Asl digit dataset," https://www.kaggle.com/datasets/hridoy1000/asl-digit, accessed: August 10, 2023.
44. M. S. Alom, M. J. Hasan, and M. F. Wahid, "Digit recognition in sign language based on convolutional neural network and support vector machine," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE, 2019, pp. 1–5.
45. M. A. Kalam, M. N. I. Mondal, and B. Ahmed, "Rotation independent digit recognition in sign language," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019, pp. 1–5.
46. V. Bheda and D. Radpour, "Using deep convolutional networks for gesture recognition in american sign language," *arXiv preprint arXiv:1710.06836*, 2017.
47. M. Neff, I. Albrecht, and H.-P. Seidel, "Gesture modeling and animation by imitation," in *ACM SIGGRAPH 2008 classes*, 2008, pp. 1–64.
48. P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 1–7.
49. P. Narayana, J. R. Beveridge, and B. A. Draper, "Gesture recognition: Focus on the hands," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5235–5244.
50. D. Konstantinidis, C. Tzelepis, D. Galanopoulos, P. Daras, and A. Iosifidis, "Sign language recognition using pose-based convolutional neural networks," *Pattern Recognition Letters*, vol. 153, pp. 20–27, 2022.