American Sign Language Character Recognition using Convolutional Neural Networks

Atesam Abdullah
Faculty of Computer Science & Engg.
GIK Institute of Engg. Sciences & Tech.
Topi, Khyber Pakhtunkhwa, Pakistan.
atesamabdullah8@gmail.com
Github

Nisar Ali
Faculty of Engineering and Applied Science
University of Regina
Regina, Canada
nay095@uregina.ca

Raja Hashim Ali
Faculty of Computer Science & Engg.
AI Research Group
GIK Institute of Engg. Sciences & Tech.
Topi, Khyber Pakhtunkhwa, Pakistan.
hashim.ali@giki.edu.pk

Zain ul Abideen
Faculty of Computer Science & Engg.
AI Research Group
GIK Institute of Engg. Sciences & Tech.
Topi, Khyber Pakhtunkhwa, Pakistan.
zain-ul-abideen@giki.edu.pk

Ali Zeeshan Ijaz
Faculty of Computer Science & Engg.
AI Research Group
GIK Institute of Engg. Sciences & Tech.
Topi, Khyber Pakhtunkhwa, Pakistan.
ali.zeeshan@giki.edu.pk

Abdul Bais
Faculty of Engineering and Applied Science
University of Regina
Regina, Canada
abdul.bais@uregina.ca

***Abstract*—This study presents a convolutional neural network (CNN) architecture developed using the TensorFlow framework to accurately recognize individual letters of American Sign Language (ASL). The CNN architecture consists of various layers including two-dimensional convolutional layers, max-pooling layers, batch normalization layers, dropout layers, and fully connected layers. The model achieved a mean validation accuracy of 95.48% and a test accuracy of 99.77% in identifying ASL characters. However, the live visual depictions revealed certain difficulties encountered by the model in identifying some ASL letters, highlighting the need for further improvement of the model's framework and dataset curation. This research contributes to the scholarly discussion on the use of machine learning approaches in identifying sign language alphabets and provides insights into the feasibility and effectiveness of utilizing these techniques in ASL recognition tasks.**

***Index Terms*—American Sign Language (ASL), Convolutional Neural Networks (CNNs), Recognition, Deep learning.**

I. Introduction

The field of computer vision has made remarkable strides in identifying and interpreting American Sign Language (ASL) using Convolutional Neural Networks (CNNs) and Tensorflow [1]–[3]. ASL serves as the primary mode of communication for the deaf community in the United States [4], and enhancing ASL translation technology can play a pivotal role in bridging communication gaps between hearing and deaf individuals. Such advancements hold great potential for promoting inclusivity in diverse settings like education, healthcare, and business [5]–[8].

Existing research has largely focused on hand-crafted methods of feature extraction and classification, but the recent progress in deep learning and CNNs has shown promising improvements in accuracy and efficiency [9], [10]. In this study, we aim to explore the use of CNN and Tensorflow to facilitate ASL translation and evaluate its effectiveness in bridging communication barriers between the hearing and deaf communities [11].

Notwithstanding the advancements in ASL translation technology, certain knowledge gaps still persist, such as the limited availability of openly accessible datasets that accurately capture the subtleties of ASL [12] and the challenges involved in translating ASL syntax and grammar into written and spoken forms of communication. Addressing these gaps is essential for further enhancing ASL translation systems and ensuring seamless and accurate communication between diverse language users. This paper aims to address the following research inquiries:

1) What is the level of precision demonstrated by the Tensorflow CNN model in identifying ASL characters? (primary inquiry)
2) To what extent does the model effectively perform in its ability to recognize characters depicted in a live video setting? (sub-inquiry)

A. Contributions and Novelty

- This study presents the design and execution of a CNN using the TensorFlow framework for the accurate recognition of characters from a specific sign language. The proposed CNN architecture includes two-dimensional convolutional layers, max-pooling layers, batch normalization layers, dropout layers, and fully connected layers, which contribute to achieving high accuracy rates.
- The research empirically validates the efficacy of deep learning architectures in addressing the challenges of sign language recognition. The model achieved a mean validation accuracy of 95.48% and a test accuracy of

99.77% in identifying characters from the targeted sign language.

- The study identifies specific characters or symbols that pose challenges for the model, such as A and M in ASL. This highlights the need for further improvement in the model's architecture and dataset curation to overcome these recognition difficulties.
- The investigation explores the potential of incorporating hand landmark data obtained through Google Mediapipe as supplementary information to enhance the model's precision and reliability in detecting intricate hand poses and gestures.
- The findings offer valuable insights into the practical applicability of the proposed model, particularly in real-time scenarios such as dynamic modification of textual content through letter addition or deletion, as well as detecting instances where no gesture is executed.

The present study details our involvement in developing a Tensorflow CNN model for ASL translation. Specifically, our contributions encompassed the implementation of the said model and the subsequent evaluation of its accuracy concerning the recognition and translation of ASL characters. The outcomes of our study offer valuable perspectives on the efficacy of Tensorflow CNN models for ASL translation. Moreover, our findings pinpoint domains that necessitate further exploration and investigation in this domain.

II. Problem Statement

This research focuses on the difficulty of accurately identifying characters or symbols from a particular sign language using deep learning techniques like CNNs. Although these models generally achieve high accuracy rates, they struggle to precisely recognize some characters or symbols. This issue limits their usefulness in practical situations. Overcoming these limitations is crucial for advancing sign language recognition technology and enabling effective communication for individuals who depend on sign language.

III. Methodology

In this study, we aimed to increase the accuracy of predicting the ASL alphabet by implementing a Convolutional Neural Network (CNN). To train and evaluate our CNN model, we utilized the ASL dataset available on Kaggle, which consists of 87,000 images measuring 200x200 pixels. The dataset comprises 29 distinct classes, including 26 classes representing the letters A-Z and three additional classes for SPACE, DELETE, and NOTHING. These specific classes allow for the deletion of a letter and the insertion of a space, which is useful for applications that require real-time classification. Furthermore, the model can accurately identify instances where no gestures are being performed by utilizing the NOTHING class.

In this study, we used a systematic approach to train a model effectively. We started by adjusting the image size to 32 by 32 pixels to ensure consistency across the dataset. Then, we normalized the images to center the input data around zero, which helped with the optimization process. After that, we used categorical encoding to improve the model's ability to understand the training data.

Next, we carefully selected hyperparameters such as the learning rate, batch size, and number of epochs through experimentation and validation. We used these hyperparameters to configure the model's architecture.

During training, we applied the model to the training dataset and used the specified hyperparameters to optimize it. We continued to iterate until the model achieved convergence, meaning the parameters didn't change significantly when trained on additional data or reached a predetermined stopping criterion.

Finally, we assessed the model's performance using a suitable metric to evaluate its accuracy or loss. We used this investigation to determine the model's proficiency in forecasting unobserved data. You can see our methodology represented in the flow diagram in Figure 2.

The parameters for our model can be found in Table I, while the model configuration is shown in Table II for this project. Our CNN architecture consists of multiple layers that process input data in a consecutive manner. The first layer is a two-dimensional convolutional layer that uses filters to learn and identify unique characteristics in the input data. In this scenario, the filters have dimensions of 3x3, resulting in 64 output channels.

The output of the convolutional layer is then passed through a max pooling layer, which reduces the spatial dimensions of the output by half, making the model more efficient. After max pooling, a batch normalization layer is applied to normalize the layer outputs and mitigate the internal covariate shift phenomenon. This helps to improve the efficiency of training deep neural networks.

The next layer is another 2D convolutional layer with 128 output channels, followed by a max pooling and batch normalization layer. A dropout layer is then applied to prevent overfitting. The final layers consist of a 2D convolutional layer with 256 output channels, followed by a max pooling and batch normalization layer. The output is then converted to a unidimensional vector and passed through a dropout layer and two fully connected layers, resulting in outputs of 1024 and 29 nodes, respectively.

The first fully connected layer has 1024 output nodes and uses the rectified linear activation function (ReLU) to improve the learning performance of deep models. The final layer is a fully connected layer with 29 output nodes, each corresponding to a specific category in the dataset. The softmax activation function is used to generate a probability distribution that assigns likelihoods to each category.

IV. Results

The TensorFlow CNN model demonstrated a mean validation accuracy of 95.48% after 15 epochs in accurately identifying ASL characters. The findings suggest that Figure 3 depicts the accuracy exhibited during training, while Figure 4 ultimately showcases the model loss during such training sessions.

Fig. 1. A screenshot of captured data [13].

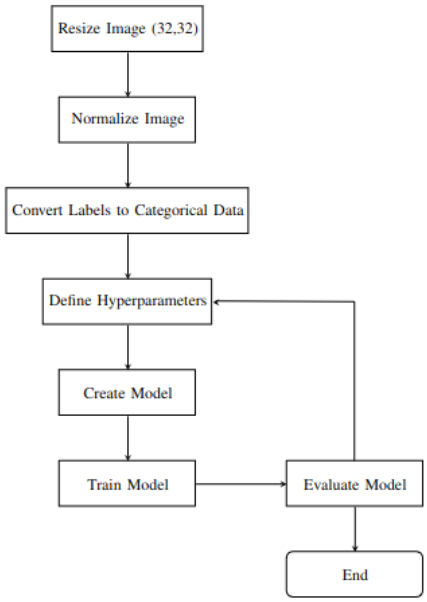

Fig. 2. Flow diagram of model creation.

TABLE I
THE PARAMETERS OF CNN USED IN THIS STUDY.

Parameters	
Epochs	15
Learning rate	0.001
Mini batch size	32
Optimizer	adam
Samples in training set	78300
Samples in validation set	8700

The confusion matrix depicting the forecasts generated by the model on the testing dataset is presented in Figure 5. As

TABLE II
THE CONFIGURATION BY LAYER OF CNN USED IN THIS STUDY.

Layer (type)	Output Shape	Param
Conv2D	(None, 32, 32, 64)	640
MaxPooling2D	(None, 16, 16, 64)	0
BatchNormalization	(None, 16, 16, 64)	256
Conv2D-1	(None, 16, 16, 128)	73856
MaxPooling2D-1	(None, 8, 8, 128)	0
BatchNormalization-1	(None, 8, 8, 128)	512
Dropout	(None, 8, 8, 128)	0
Conv2D-2	(None, 8, 8, 256)	295168
MaxPooling2D-2	(None, 4, 4, 256)	0
BatchNormalization-2	(None, 4, 4, 256)	1024
Flatten	(None, 4096)	0
Dropout-1	(None, 4096)	0
Dense	(None, 1024)	4195328
Dense-1	(None, 29)	29725

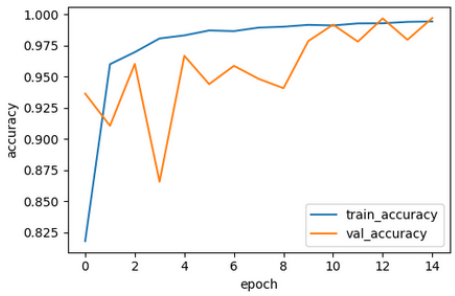

Fig. 3. Plot showing accuracy VS epochs.

evidenced by the figure, the model exhibits a commendable level of accuracy in character recognition, with the exception of instances involving similar letter forms such as A and M at edges.

In order to assess the efficacy of the model in accurately translating ASL characters, we conducted an evaluation of the

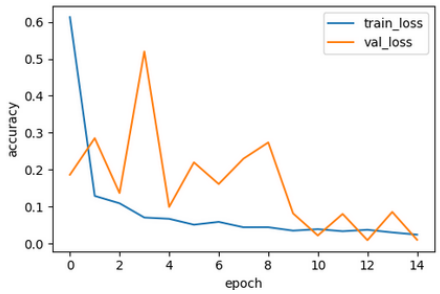

Fig. 4. Plot showing loss VS epochs.

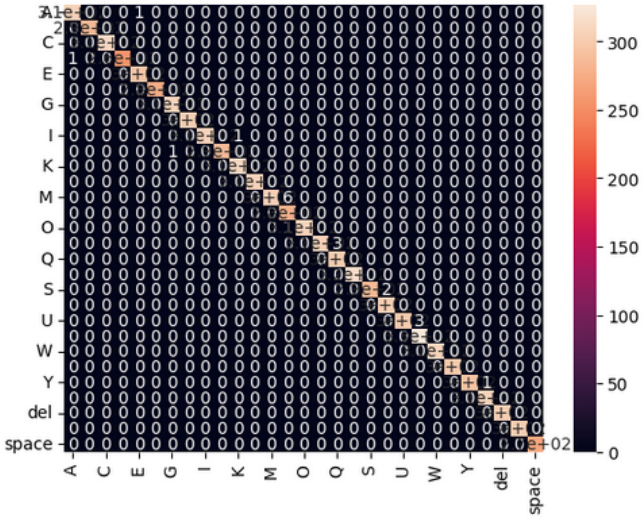

Fig. 5. Confusion matrix of all characters.

model's performance on a particular subset of the testing set comprising ASL phrases and sentences. The results of our study indicate that our model has achieved a test accuracy of 99.77% in translating ASL characters.

To investigate the impact of utilizing distinct datasets for training and testing purposes, the present study sought to assess the precision of our model on web-based imagery. Figure 6 illustrates exemplary instances of accurate and erroneous translations generated by our model.

V. Discussion

The present study reports the successful implementation of a CNN and a TensorFlow model for recognizing ASL alphabets. Specifically, the models achieved a respectable mean validation accuracy of 95.48% (in 15 epochs) and a robust test accuracy of 99.77%. The results demonstrate that deep learning architectures have considerable potential for addressing ASL recognition challenges. Nevertheless, the live images (refer to Figure 6) reveal that our model encounters difficulties in accurately recognizing particular ASL letters, notably the letters A and M. Consequently, it substantiates the urgency of refining the model architecture and dataset selection to enhance its performance.

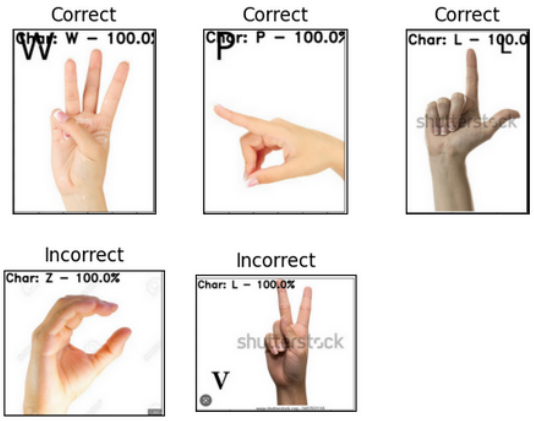

Fig. 6. Examples of correct and incorrect characters recognition [14].

TABLE III
Comparison of our approach vs other methods.

Method	Pros	Cons
Our CNN Model	- Utilizes RGB images which capture more visual information than grayscale images - Able to learn more complex features through multiple convolutional layers - Achieved high accuracy on our ASL alphabet classification task	- Requires a large amount of labeled data to train effectively - Computationally intensive and requires high-end hardware to train in a reasonable time frame
Traditional Machine Learning	- Requires less data to train effectively - Generally faster to train and less computationally intensive than deep learning models	- Limited by feature engineering, which may be difficult to do on image data - May not perform as well on complex tasks with large amounts of data
Rule-Based Approach	- Simple and easy to implement - Does not require large amounts of data or computational power	- Limited by the ability of the rule set to capture all variations in hand gestures - May not perform as well on complex tasks with large amounts of data

The focus of our study concerns the investigation of deep learning models for the purpose of performing ASL alphabet recognition tasks. Specifically, our approach aims to leverage the CNN architecture and the TensorFlow framework to achieve this objective. By conducting this research, we aim to contribute to the academic discourse surrounding the application of machine learning techniques to the recognition of sign language alphabets. The findings of this study provide empirical evidence exhibiting the practicability and efficacy of employing these methodologies in tasks concerning ASL recognition.

A feasible enhancement for our ASL recognition model could entail using hand landmark information obtained through image extraction via Google Mediapipe. By utilizing this supplementary data, there is a possibility that the dependence of the model solely on identifying patterns based on the physical characteristics of the hand can be lessened. This

would enable the model to concentrate on the fundamental movements and positions of the hand.

Moreover, the inclusion of hand landmark data has the potential to enhance the model's efficacy in detecting intricate hand poses and gestures that are commonly difficult to recognize through conventional image-based methods accurately. Additional investigation in this direction may result in noteworthy progressions in the domain of ASL recognition technology, thereby enhancing its accessibility and efficacy for individuals who are deaf or hard of hearing. Prospective investigations may prioritize the enhancement of the model's architecture for the purpose of mitigating the difficulties presented by specific ASL letters. Moreover, there is potential to enhance the generalizability and accuracy of the model by delving into more diverse and extensive datasets.

Our research offers evidence supporting the efficacy of deep learning models in the context of ASL alphabet recognition tasks. There exists potential for forthcoming investigations to concentrate on advancing the model architecture whilst also delving into the utilization of hand landmark data with a view to augmenting precision and resilience. The present investigation lays the groundwork for future progressions in ASL recognition technology, enhancing its accessibility and efficacy in practical situations.

VI. Conclusion

In this study, we used a CNN to recognize the letters of the ASL alphabet. Our data were obtained from the publicly available ASL dataset on Kaggle, which included 29 categories, including 26 letters from A to Z, as well as SPACE, DELETE, and NOTHING. Our methodology involved resizing the images to 32 by 32 pixels and normalizing them. We established specific hyperparameters, such as batch size, number of epochs, and learning rate, to construct our CNN model. Our model achieved an impressive mean validation accuracy of 95.48% (over 15 epochs) and a high accuracy of 99.77% on the test set for ASL alphabet recognition. These results demonstrate the effectiveness of deep learning architectures for ASL recognition tasks. However, our model encountered difficulties in accurately identifying certain letters, such as A and M, indicating the need to improve the model's architecture and dataset selection or transition to using landmark data. Our study aims to improve the accuracy and efficiency of ASL alphabet recognition tasks using deep learning models implemented with the CNN architecture and the TensorFlow framework.

References

[1] D. Sulfayanti and L. Armin, "Sign language recognition using modified convolutional neural network model," in *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, 2018, pp. 1–5.

[2] S. He, "Research of a sign language translation system based on deep learning," in *2019 International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*. IEEE, 2019, pp. 392–396.

[3] N. Takayama and H. Takahashi, "Data augmentation using feature interpolation of individual words for compound word recognition of sign language," in *2020 International Conference on Cyberworlds (CW)*, 2020, pp. 137–140.

[4] N. Ali, Z. Halim, and S. F. Hussain, "An artificial intelligence-based framework for data-driven categorization of computer scientists: a case study of world's top 10 computing departments," *Scientometrics*, vol. 128, no. 3, pp. 1513–1545, 2023.

[5] A. Z. Ijaz, R. H. Ali, N. Ali, T. Laique, and T. A. Khan, "Solving graph coloring problem via graph neural network (gnn)," in *2022 17th International Conference on Emerging Technologies (ICET)*. IEEE, 2022, pp. 178–183.

[6] A. A. Khan, R. H. Ali, and B. Mirza, "Evolutionary history of alzheimer disease-causing protein family presenilins with pathological implications," *Journal of Molecular Evolution*, vol. 88, no. 8-9, pp. 674–688, 2020.

[7] D. K. Mohammad, R. H. Ali, J. J. Turunen, B. F. Nore, and C. E. Smith, "B cell receptor activation predominantly regulates akt-mtorc1/2 substrates functionally related to rna processing," *PloS one*, vol. 11, no. 8, p. e0160255, 2016.

[8] K. Dabre and S. Dholay, "Machine learning model for sign language interpretation using webcam images," in *Proceedings of 2014 International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2014.

[9] M. Sincan, O. Tur, A. O, Y. Keles, and H, "Isolated sign language recognition with multi-scale features using lstm," in *27th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2019, pp. 1–4.

[10] Y. Yang, S. Tu, R. H. Ali, H. Alasmary, M. Waqas, and M. N. Amjad, "Intrusion detection based on bidirectional long short-term memory with attention mechanism," *CMC – Computer Material and Continua*, vol. 74, no. 1, pp. 1597–1632, 2022.

[11] D. Pahuja and S. Jain, "Recognition of sign language symbols using templates," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. IEEE, 2020, pp. 1157–1160.

[12] Y. Patil, S. Krishnadas, A. Kastwar, and S. Kulkarni, "Ai-enabled real-time sign language translator," in *Soft Computing: Theories and Applications*. Springer, 2020, pp. 1–15.

[13] Akash, "Asl alphabet," Apr 2018. [Online]. Available: https://www.kaggle.com/datasets/grassknoted/asl-alphabet

[14] [Online]. Available: https://www.shutterstock.com/search/sign-language-alphabet