Computer Vision Based Bangla Sign Language Recognition Using Transfer Learning

Md Rezwane Sadik
University of South Dakota
Vermillion, SD 57069, USA
mdrezwane.sadik@coyotes.usd.edu

Rayhanul Islam Sony
Trine Univesity
Angola, IN 46703, USA
rsony23@my.trine.edu

Nuzhat Noor Islam Prova
Pace University, Seidenberg School of CSIS
One Pace Plaza, New York City, NY 10038, USA
nuzhatnsu@gmail.com

Md Masum Rana
University of South Dakota
Vermillion, SD 57069, USA
hmdmasum.rana@coyotes.usd.edu

Md.Shariful Islam
Department of Computer Science & Engineering
Bangladesh University of Business & Technology
Mirpur,Dhaka , Bangladesh
sisijan4@gmail.com

Sinhad Hossain Fahim
Department of Computer Science & Engineering
Stamford University Bangladesh (SUB)
Dhaka, Bangladesh
fahim1310652538@gmail.com

Abdullah Al Maruf*
Department of Computer Science & Engineering
Bangladesh University of Business & Technology
Mirpur, Dhaka, Bangladesh
marufbubt822@gmail.com

***Abstract*—Human societies have relied on communication since ancient times, yet verbal communication poses significant challenges for deaf and hard-of-hearing individuals, necessitating reliance on sign language. Recent advancements, notably through deep learning models, have propelled research in this area. Acknowledging the need for further progress, particularly in minority languages like Bengali, our study aims to develop a method for image-based Bengali sign language detection. We constructed two independent convolutional neural network (CNN) models, InceptionV3 and Xception, leveraging data from diverse sources. Remarkably, the Inception V3 model achieved an accuracy of 97%, while the Xception model surpassed expectations with an accuracy of 99.50%. These results signify substantial progress, demonstrating the efficacy of deep learning architectures, especially the Xception model, in accurately interpreting Bengali sign language. Our study demonstrated how transfer learning, when combined with careful optimization, may yield remarkable outcomes in Bengali sign language recognition, which are further enhanced by data augmentation methods.**

***Index Terms*—Bengali Sign Language, Deep Learning, BDSL, Image Classification, Data Augmentation**

I. Introduction

Hearing loss or impairment is represented by a partial or comprehensive inability to hear, and it can be in either one ear or both [1]. According to data from 2013, over 1.1 billion individuals were involved in hearing loss, and this condition led to a total inability to hear in around 538 million people, resulting in moderate to extreme disabilities for almost 124 million individuals [2]. Individuals with hearing and speech impairments, generally referred to as Deaf and Dumb (D&D), depend on sign language as their immediate mode of communication to express emotions and thoughts. Social activities can be challenging for D&D individuals due to the broad population's limited understanding of sign language. Only a small number of individuals outside the D&D community can understand sign language, and there is a big communication gap between D&D individuals and the general public. D&D individuals need help understanding spoken language, and lip reading is usually not a viable option for them [3].

Sign language recognition technologies play a vital role in identifying different gestures, including numbers, alphabets, phrases, and distinctive signs that are operated for traffic signals [4]. Nearly every spoken language globally is escorted by its corresponding verified sign language. Language acts as the primary standard of human communication, containing spoken, written, and symbolic forms, and sign language dictionaries adeptly define the foundational elements and gestures of sign languages. Real-world applications raise challenges due to pronounced differences at phonological, morphological, grammatical, and lexical levels. Various sign languages have appeared globally, including American Sign Language (ASL), Indian sign languages [5], Arabic sign language [6], French Sign Language, British Sign Language (BSL), and Japanese Sign Language (JSL), among others. Notably, these languages developed independently, and Bangladeshi Sign Language (BdSL) stands different from the rest.

Researchers are separated focus, with some emphasizing real-time symbol recognition while others focus on static images. In contemporary developments, artificial neural network-based approaches have been used for real-time recognition of

*Corresponding author

979-8-3503-8166-5/24/$31.00 ©2024 IEEE

ASL terms and alphabets [7]. Researchers use different techniques, such as neural networks (NN), Principal Component Analysis (PCA), Hidden Markov Model (HMM), and fuzzy logic [8], to facilitate sign language recognition. Moreover, Deep Learning (DL) and Machine Learning (ML) techniques have successfully recognized signs in diverse sign languages. These advancements emphasize the versatility of sign language recognition technologies in various linguistic and cultural contexts.

In our research, we have applied deep transfer learning methods with data analysis techniques to detect Bengali sign language. The contribution of our research :

- We used transfer learning to improve performance in Bengali sign language detection by fine-tuning multiple pre-trained models and utilizing their knowledge from large-scale datasets.
- In our research, we have overcome dataset scarcity by using different data augmentation techniques .
- Our proposed model demonstrated robust and generalized performance, characterized by the absence of overfitting or underfitting issues, ensuring its reliability across diverse datasets and scenarios.
- Our proposed model achieved remarkable accuracy while outperforming the previous modeling in all evaluation criteria.

The paper is organized as follows: In Section II, we will present a literature review, which carefully reviews previous studies on sign language. In Section III, we will present our proposed method and research procedures. We have presented our research outcomes in Section IV, where we will demonstrate the efficacy of our method. Finally, the conclusion of the research is presented in Section V.

II. Related Work

Researchers have used various methodologies and procedures to develop a Bengali sign language detection system. In this section, we have described some of the existing research works in the literature.

Hossen et al. [9] recognized the importance of using DCNN to recognize 37 Bengali sign languages by conducting an experiment that involved analyzing 1147 images collected from here [10]. To optimize the performance of a pre-trained VGG16 network that had already been trained on the large ImageNet dataset, the researchers carefully employed transfer learning and fine-tuning techniques. Although the suggested approach was innovative and performed well during the training phase with an accuracy rate of 96.33%, it did not perform as well on the testing set, with an accuracy of 84.68%.

A comprehensive compilation of 12,581 unique hand signals representing the 38 alphabets of BdSL was created in collaboration with the National Federation of the Deaf. To solve the complex problem of recognition among these numerous classes, researchers [11] proposed using a convolutional neural network(CNN) based on VGG19. The purpose of this neural architecture was to achieve a comprehensive approach to categorization by recognizing even the minute details of every hand gesture. Their efforts resulted in an impressive overall test accuracy of 89.6%, demonstrating the effectiveness of their proposed method for sign language recognition. However, they didn't evaluate their model using other's evaluation criteria.

Authors [12] aimed to recognize nine different Bengali sign languages in real time using a CNN model. The research began with a small dataset of 160 images, which were converted to binary format for training. The author introduced an innovative technique of systematically rotating the images, resulting in significant performance improvements. The accuracy of the system improved remarkably from 94.17% before rotation to an impressive 99.75% after deployment. This novel approach to image alteration played a game-changing role in the study's findings.

Various research has been conducted on smaller datasets with high accuracy. However, author [13] proposed a segmentation, augmentation, and convolutional CNN-based classification approach to evaluate three benchmark datasets: "38 BdSL," "KU-BdSL," and "Ishara-Lipi." During the segmentation step, YCbCr, HSV, and a watershed algorithm are used together to detect gesture indications precisely. Finally, the CNN-based model known as BenSignNet is used to extract features and classify them. Surprisingly, the suggested technique achieves accuracies of 94.00%, KU-BdSL, and Ishara-Lipi, respectively, and 99.60% overall accuracy. Compared to traditional methods, the experimental findings validate the higher identification rates achieved by the suggested strategy, highlighting its ability to generalize across various BSL datasets.

Author [14] focused on recognizing Bengali alphabetic sign language through hand gestures. It analyzes a set of 36 Bengali alphabets and tests a suggested architecture on the "Ishara Lipi" dataset. This dataset comprises 3600 photos that correspond to 1800 Bengali characters. The CNN architecture applied in the study achieves an impressive precision level of 99.86% despite the use of a small dataset. This achievement marks a significant advance in the field of Bengali alphabet recognition, surpassing previous attempts.

The application of multiple transfer learning models to identify multiclass static sign language words for the deaf and dumb community in Bangladesh was investigated by Islam et al. [15]. They used four distinct models: VGG16, VGG19, AlexNet, and InceptionV3. They used a dataset with 1105 images. The models demonstrated excellent training accuracies with 99.92% for VGG16, 99.58% for VGG19, 98.70% for InceptionV3, and 97.86% for AlexNet. Though the validation accuracy was lower, VGG16 did the best, scoring 92.41% [15]. Das et al. [16] proposed a six-layer ConvNeural network designed to recognize numerals and alphabets in sign language. Their approach concentrated on thorough preprocessing and data augmentation to improve model performance. Their system remarkably demonstrated the usefulness of their architectural decisions early in the training process, achieving a training accuracy of over 90% [16]. Nihal et al. [17] enhanced the recognition of Bangla Sign Language by combining CNN architectures with Zero-Shot Learning. This combination worked well, achieving an impressive 93.68% accuracy rate when processing a dataset of 35,149 photos. During the training phase, this technique was especially good at identifying indications that weren't previously seen [17].To address Bangla Sign Language recognition, Poddar et al. [18] used a DenseNet201 architecture inside a feature pyramid network. One of the best-performing models in their investigation, their model had a remarkable accuracy of 98.644% when tested against a variety of difficult backdrops [18]. A thorough deep-learning system for sophisticated sign language recognition was created by Rahaman et al. [19]. Their model achieved an accuracy of 98.38% on their testing set as a result of their thorough training with advanced image processing techniques. This work emphasizes how deep learning can improve sign language recognition systems' accuracy [19].

III. Methodology

In this section, we have presented our suggested approaches and methods. Detailed descriptions of all steps are described here. The overall framework of our research is shown graphically in Figure 1.

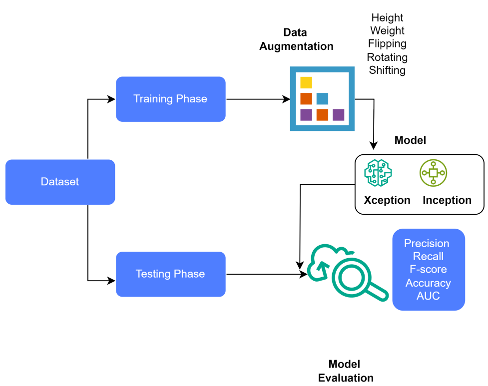

Fig. 1. Overall Framework of our research

A. Dataset Description and preprocessing

The dataset contains 1075 images. The dataset has 10 classes. The original dataset contains 36 classes with 1800 images [20]. The number of classes in the dataset needed to be more balanced, and instead of focusing on the whole dataset, we have particularly worked on 10 classes. So, Firstly, we divided the dataset for training and validation. The 925 images belonged to 10 classes for training, and 150 images belonged to 10 classes for validation. Then, we tried a data augmentation technique in the training dataset. The sample dataset is shown in Figure 2.A thorough preparation process was applied to the original dataset, which included tagging, image cropping, grayscale conversion, and scaling. The resizing process, in particular, required a conversion from the initial 128x128 measurements to our revised 299x299 standards. This change was limited to changing the values of the pixels, making sure that they matched the parameters of our model.As there are not many images in the dataset, we intended to increase the images artificially. Overfitting is prevented by data augmentation

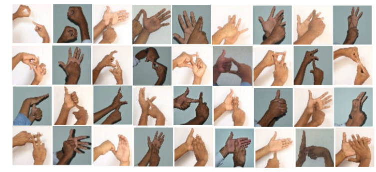

Fig. 2. Example of Sample Images in the Dataset

techniques, and learning algorithms perform better. Getting a real-time dataset to feed a model for training is a complex and time-consuming process. By applying different changes or alterations to the preexisting data, data augmentation is a deep learning technique that creates new training examples, artificially increasing the size and diversity of a dataset. It is frequently employed when the dataset that is supplied is little or unbalanced or when the model must be more resilient to changes in the input data. Since there isn't much data available, we used data augmentation strategies in our study. Applying rotation, scaling, cropping, or adding noise to the original data can be used in data augmentation to create new samples that capture various variations and patterns. It

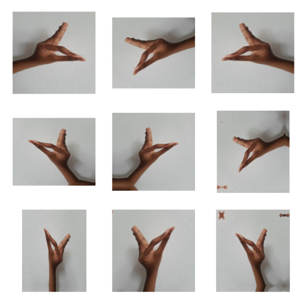

Fig. 3. Sample of an Image of a Specific Data After Data Augmentation

is frequently employed when the dataset that is supplied is little or unbalanced or when the model must be more resilient to changes in the input data. Since there isn't much data available, we used data augmentation strategies in our study. Applying rotation, scaling, cropping, or adding noise to the original data can be used in data augmentation to create new samples that capture various variations and patterns. In the current research, we applied Random Horizontal Flip, Random Rotation, Random Zoom, Random Height Shift, and Random Width Shift [21]. The samples of the dataset after augmentation are shown in Figure 3. The figures show that some figure heights are greater than others (height and weight increased), and some flipped and rotated.

B. Proposed Model

In our research, we have applied Inceptionv3 and Xception Transfer learning models.We have fine-tuned the models according to our research by adding additional layers and putting different values. The general architecture of the proposed transfer learning model (Inception, Xception) is shown in Figure 4.

Inceptionv3 : Inception modules are the essential building blocks of the Inceptionv3 model, a CNN architecture designed for image classification tasks. These modules are made up of pooling operations and parallel convolutional paths with different filter sizes (1x1, 3x3, and 5x5). The outputs of these paths are concatenated to improve feature extraction. The output of every inception module can be expressed mathematically as follows:

$$\text{Concatenate}(\text{Conv}_{1x1}(X), \text{Conv}_{3x3}(X), \text{Conv}_{5x5}(X), \text{Pooling}(X))$$

in which the input feature maps are indicated by X, and the operations performed within each path are represented by Conv_{1x1}, Conv_{3x3}, Conv_{5x5}, and Pooling. The pre-trained On ImageNet InceptionV3 model's power is expertly combined with well-planned alterations in the architectural blueprint for any image classification tasks [22]. We have customized the inception model according to our research. Layers up to index 290 are diligently frozen in order to protect the inherent knowledge that the InceptionV3 captures. This helps the model hold onto its learned features for later training cycles [23]. The input layer is carefully constructed to fit the desired image shape, allowing for the smooth incorporation of external data augmentation using a custom function intended to improve the flexibility and robustness of the model. Subsequently, the input images undergo an intelligent preparation step using the unique preprocessing function of the InceptionV3. In order to mitigate the vanishing gradient problem during training, auxiliary classifiers are incorporated into the network and placed in strategic locations to offer extra supervision. The next layer highlights the model's capacity to recognize complex patterns and subtleties by engaging the preprocessed images in an elaborate dance of feature extraction. This is followed by global average pooling, a sophisticated process that extracts crucial spatial information and enhances the model's ability to capture meaningful representations. To produce class probabilities, these classifiers employ fully connected layers after global average pooling. The auxiliary classifiers' output can be expressed mathematically as follows:

$$Y_{\text{aux}} = \text{Softmax}(\text{GlobalAveragePooling}(X) \times W + b)$$

where X stands for the feature maps, W and b for the classifier layer's weights and biases, and Y_{aux} for the output of the auxiliary classifier. A carefully inserted dropout layer, with a discriminating dropout rate of 0.2, modifies the model's ability to generalize in order to reduce the likelihood of overfitting. The final classifier at the end of the network, which consists of fully connected layers and a softmax activation function, produces the final classification predictions. The final classifier's mathematical output looks like this:

$$Y_{\text{final}} = \text{Softmax}(\text{Flatten}(X) \times W + b)$$

Where the feature maps are represented by X, the classifier layer's weights and biases are indicated by W and b, and the final classification output is shown by Y_{final}.

Xception : The Xception structure is an expanded version of

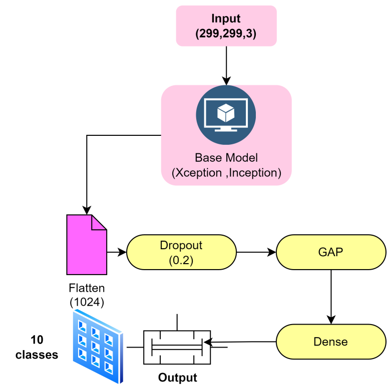

Fig. 4. Proposed transfer learning model architecture

TABLE I
PARAMETERS OF THE MODELS DURING COMPILATION AND TRAINING

Model	Description
Xception	Optimizer: keras.optimizers.adam ,Learning_rate :1e-2 Decay_steps : 10000, Decay_rate :0.9 loss= :categorical_crossentropy metrics accuracy ,epochs : 300
Inceptionv3	Optimizer: tf.keras.optimizers.adam ,Learning_rate :0.001 loss= :tf.keras.losses.SparseCategoricalCrossentropy metrics accuracy,epochs : 100

the Inception layout that uses depthwise discrete convolutions in place of the original Inception units. It is the pointwise convolution with the depthwise convolution coming before it. Furthermore, the residual/skip links similar to the ResNet design are included in this Xception architecture [24]. In this research, We are utilizing the Xception model, an advanced convolutional neural network that is well-known for its effectiveness in tasks involving images. We seek to leverage the power of the Xception model's learned hierarchical features while customizing it to our particular classification requirements by starting it with pre-trained weights from ImageNet, hence excluding its classifier or fully connected layers [25]. In our research, the input shape is set to (299, 299, 3). We inserted a custom classification head since the layers that come after, excluding the original classifier, are flattened. The layers that make up this head are closely spaced and have activation functions for rectified linear units (ReLUs). This is a purposeful design decision since ReLU adds non-linearity, which helps the model recognize complex patterns in the data. The last layer, which has ten neurons and a softmax activation function, indicates the output requirements for our particular task, which suggests that it is a classification problem with ten different classes.

Both of the model is compiled using several parameters. The model compilation processes and training process parameters are shown in Table I.

IV. RESULT & DISCUSSION

Our study was conducted with Google Colab Pro+, which has very low system requirements. But in terms of software requirements, we used libraries like TensorFlow, Keras, NumPy, sci-kit-learn, and OpenCV, which are all easily accessible in the Google Colab environment, along with Python 3.7 or higher. Additionally, faster model training and evaluation were made possible by utilizing GPU support through Google Colab's integrated GPU acceleration feature (T4 GPU with High RAM). Even though Google Colab does not require any particular hardware specifications, we still recognize that it is crucial to record the software configurations and dependencies in order to guarantee reproducibility and make it easier for other researchers to carry out similar studies. We have evaluated our models using several evaluation criteria. We have calculated precision, recall,f-score, accuracy, and confusion matrix also will show the training and validation loss and training and validation accuracy of the best model. At the end, we will present the class-wise precision, recall, and f-score of the Xception model because, in our research Xception model performed best.

The performance of both models is shown in Table II. The presented performance metrics for Xception and InceptionV3, namely precision, recall, and F-score, deliver insights into the effectiveness of these image classification models. Xception stands out with a remarkable precision of 97%, demonstrating a high level of accuracy when predicting positive instances and indicating that when Xception asserts an image belongs to a particular class, it is correct 97% of the time. In comparison, InceptionV3, while still acquiring a commendable precision of 92%, lags slightly behind Xception in terms of minimizing false positives and offers that Xception shows a higher level of discernment in its positive predictions. Moving on to recall, Xception gains 96%, signifying its capability to capture a considerable percentage of actual positive instances and implying that Xception excels in determining the majority of relevant instances in the dataset. On the other hand, InceptionV3, with a recall of 91%, is marginally less influential in capturing all positive instances and indicating that Xception displays a higher sensitivity. Considering the F-score, Xception holds an impressive 96%, and a balanced score shows that Xception excels in both precision and recall, reaching a harmonious trade-off between the two. In contrast, InceptionV3's F-score of 91.50% is admirable but narrowly trails behind Xception. While InceptionV3 maintains a good balance, it might sacrifice a bit of precision or recall corresponding to Xception. In a nutshell, Xception outperforms InceptionV3 in all metrics, showcasing higher accuracy in positive predictions, more suitable coverage of true positive instances, and overall well-balanced performance. The training and validation loss graph of the Xception model is shown in Figure 5. The loss value is 0.0618.

TABLE II
PERFORMANCE OF THE PROPOSED MODELS

Model	Precision	Recall	F-score	Accuracy
Xception	97%	96%	96%	99.50%
InceptionV3	92%	91%	91.50%	97%

The training and validation accuracy curve of the Xception model is shown in Figure 6.In our research Xception model performed with 99.50 accuracy, whereas the InceptionV3 model showed 93.77% accuracy. In binary and multiclass classification, Receiver Operating Characteristic (ROC) curves are frequently used to assess a classification model's performance. In addition to ROC curves, a number of indicators are calculated to offer a thorough evaluation of the model's effectiveness. We computed the ROC for each class separately and then took the macro and micro averages to assess our model. In Figure 7, the ROC curve is displayed. The result showed that we have got 100% ROC value for every class, except class 6 and class 9. Class 6 and class 9 ROC values are accordingly 94% and 99%. The micro and macro average is 99% also. Finally, we evaluated our model by the confusion matrix. The confusion matrix details the model's performance in each class when it comes to multiclass classification, taking into consideration numerous classes. Understanding the advantages and disadvantages of a categorization model requires an interpretation of the confusion matrix. It helps determine the model's strong points and potential areas for improvement,

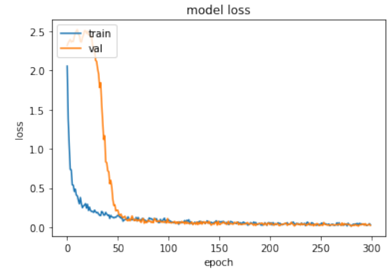

Fig. 5. Training and Validation Loss of Xception Model

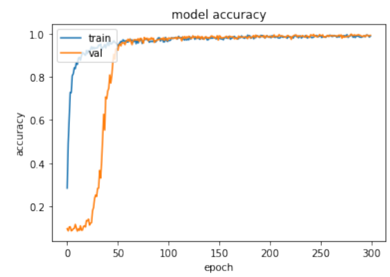

Fig. 6. Training and Validation Accuracy of Xception Model

Fig. 7. ROC Curve obtained by Xception Model

TABLE III
COMPARISON THE PROPOSED WORK WITH PRIOR RESEARCH

Ref	Dataset	Model	Accuracy
[9]	BDSL(1147 images)	DCNN	96.33%
[11]	328 samples	CNN	89.6%
[14]	Ishara-Lipi (50 sets)	CNN	99.86%
proposed model	Ishara-Lipi	Xception	99.50%

directing the model's fine-tuning for improved performance. The confusion matrix is shown in Figure 8.

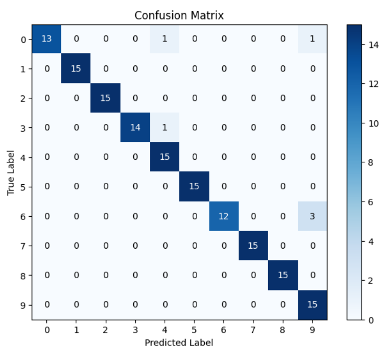

Fig. 8. Confusion Matrix

A. Discussion

We recognize that our models have limitations, including problems with biases and computational limitations. Class-specific performance metrics are revealed by our analysis, as shown in Table IV. Some classes have a high F-score of 100%, while others have a relatively low F-score, which is 79%. The observed difference in the distribution of F-scores between classes points to the possibility of bias in the classification results. There are a number of potential causes for this bias. These biases have the potential to distort the model's assessment of its overall performance and hinder its application in a variety of real-world contexts. These restrictions might make it more difficult to apply our findings in real-life situations. Moreover, we didn't use a different dataset to test our model. We worked with the Ishara-Lipi dataset, which originally had 36 classes. However, we only took into account 10 of those classes. The suggested model, however, achieved greater generalization capability and had no problems with overfitting or underfitting. We've included a comparison with other cutting-edge models or methods in the industry. We provide more context for assessing our models' efficacy and pinpointing areas for development by comparing our findings to previously published research. We have compared our proposed model with prior research works in Table III. Detecting sign language presents a significant challenge, and the need for more sufficient data is one of the major obstacles that researchers face. Many researchers rely on small-sized datasets, which limits the accuracy of their models. Rafi et al. [11] used only 328 samples in their study. They got 89.6% accuracy, whereas our proposed model achieved 99.50% accuracy. However, using the large sample compared to [11], Hossen et al. [9] achieved 96.33% accuracy using The DCNN model. They used the BDSL dataset and achieved an accuracy of 96.33%, which is lower than our proposed model. Our proposed model was trained on the Ishara Lipi dataset, a similar dataset used in a previous study [14], and achieved an impressive accuracy of 99.86%. However, it is worth noting that the accuracy alone does not determine the overall performance of the model. Other evaluation metrics such as precision, recall, and f-score are also important when the dataset is imbalanced.Owing to the Xception model's better performance on all assessed metrics, we advise deploying it first rather than InceptionV3, based on our findings and comparative analysis. But we also stress how crucial it is to take particular use cases and resource limitations into account when choosing the best model for real-world applications.

TABLE IV
PERFORMANCE METRICS CLASS-WISE OF XCEPTION MODEL

Class	Precision	Recall	F1-Score
Class 0	1.00	0.87	0.93
Class 1	1.00	1.00	1.00
Class 2	1.00	1.00	1.00
Class 3	0.93	0.93	0.93
Class 4	0.93	0.93	0.93
Class 5	0.94	1.00	0.97
Class 6	0.92	0.80	0.86
Class 7	1.00	0.93	0.97
Class 8	1.00	1.00	1.00
Class 9	0.79	1.00	0.88

In our research, we applied the data augmentation techniques where the dataset size is increased. To improve the data quality, data augmentation techniques such as shifting, weighting, and zooming were applied. These techniques ensured that individual data was captured from different angles, enabling accurate identification. So, the proposed model is able to take input from different types of images. Also, our model performance is good for all of the classes. One major thing is that our system is able to take grayscale input. Class-wise performance of the proposed Xception model is shown in Table IV.

V. CONCLUSION

Developing a sign language recognition system for the mute-deaf community has been a topic of substantial interest among researchers for a comprehensive period. We have suggested a framework for a system that employs the advances in Deep Transfer Learning to build a Bengali Sign Language Recognition system. Employing DL has greatly supported the enhancement of the performance of such a recognition system compared to models proposed in earlier papers. We have used Ishara-Lipi: Bangla Sign Language Dataset with the pre-trained network and demonstrated that our framework can perform reliable results. Through the accurate calculation of various evaluation metrics, we have systematically evaluated the performance of our models, providing a comprehensive understanding of their effectiveness in Bengali Sign Language Recognition. Notably, our Xception model appeared as the frontrunner, achieving the highest accuracy and showing remarkable proficiency in recognizing Bengali Sign Language gestures. Our framework's result emphasizes the potential of deep transfer learning to address the complexities of sign language recognition, thereby contributing to the advancement of the Deaf and Dumb (D&D) community. By concentrating on expanding class variations, enhancing computational complexity, and working with a variety of datasets, future research can pave the way for sign language recognition systems that are even more reliable and inclusive. Various signing styles and backgrounds represented in a variety of datasets would improve the system's flexibility and ability to generalize to real-world situations. Furthermore, by increasing class variations, the system would be able to recognize a wider variety of sign language gestures, meeting a wide range of user needs. Furthermore, reducing computational complexity would expedite the deployment process, improving the technology's usability and accessibility for broad application.

References

[1] M. M. Hasan, A. Y. Srizon, and M. A. M. Hasan, "Classification of bengali sign language characters by applying a novel deep convolutional neural network," in *2020 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2020, pp. 1303–1306.

[2] T. Vos, R. M. Barber, B. Bell, A. Bertozzi-Villa, S. Biryukov, I. Bolliger, F. Charlson, A. Davis, L. Degenhardt, D. Dicker *et al.*, "Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: a systematic analysis for the global burden of disease study 2013," *The lancet*, vol. 386, no. 9995, pp. 743–800, 2015.

[3] B. C. Karmokar, K. M. R. Alam, M. K. Siddiquee *et al.*, "Bangladeshi sign language recognition employing neural network ensemble," *International journal of computer applications*, vol. 58, no. 16, pp. 43–46, 2012.

[4] F. Shamrat, S. Chakraborty, M. M. Billah, M. Kabir, N. S. Shadin, and S. Sanjana, "Bangla numerical sign language recognition using convolutional neural networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 405–413, 2021.

[5] S. Das, S. K. Biswas, and B. Purkayastha, "A deep sign language recognition system for indian sign language," *Neural Computing and Applications*, vol. 35, no. 2, pp. 1469–1481, 2023.

[6] E. Aldhahri, R. Aljuhani, A. Alfaidi, B. Alshehri, H. Alwadei, N. Aljojo, A. Alshutayri, and A. Almazroi, "Arabic sign language recognition using convolutional neural network and mobilenet," *Arabian Journal for Science and Engineering*, vol. 48, no. 2, pp. 2147–2154, 2023.

[7] B. Alsharif, A. S. Altaher, A. Altaher, M. Ilyas, and E. Alalwany, "Deep learning technology to recognize american sign language alphabet," *Sensors*, vol. 23, no. 18, p. 7970, 2023.

[8] T. Ingle, S. Daware, N. Kumbhar, K. Raut, P. Waghmare, and D. Dhawase, "Sign language recognition," *Scandinavian Journal of Information Systems*, vol. 35, no. 1, pp. 294–298, 2023.

[9] M. Hossen, A. Govindaiah, S. Sultana, and A. Bhuiyan, "Bengali sign language recognition using deep convolutional neural network," in *2018 joint 7th international conference on informatics, electronics & vision (iciev) and 2018 2nd international conference on imaging, vision & pattern recognition (icIVPR)*. IEEE, 2018, pp. 369–373.

[10] K. E. Aziz, A. Wadud, S. Sultana, M. A. Hussain, and A. Bhuiyan, "Bengali sign language recognition using dynamic skin calibration and geometric hashing," in *2017 6th international conference on informatics, electronics and vision & 2017 7th international symposium in computational medical and health technology (ICIEV-ISCMHT)*. IEEE, 2017, pp. 1–5.

[11] A. M. Rafi, N. Nawal, N. S. N. Bayev, L. Nima, C. Shahnaz, and S. A. Fattah, "Image-based bengali sign language alphabet recognition for deaf and dumb community," in *2019 IEEE global humanitarian technology conference (GHTC)*. IEEE, 2019, pp. 1–7.

[12] K. Ahammad, J. A. B. Shawon, P. Chakraborty, M. J. Islam, and S. Islam, "Recognizing bengali sign language gestures for digits in real time using convolutional neural network," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 19, no. 1, 2021.

[13] A. S. M. Miah, J. Shin, M. A. M. Hasan, and M. A. Rahim, "Bensignnet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network," *Applied Sciences*, vol. 12, no. 8, p. 3933, 2022.

[14] M. J. Hossein and M. S. Ejaz, "Recognition of bengali sign language using novel deep convolutional neural network," in *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE, 2020, pp. 1–5.

[15] M. M. Islam, M. R. Uddin, M. N. AKhtar, and K. R. Alam, "Recognizing multiclass static sign language words for deaf and dumb people of bangladesh based on transfer learning techniques," *Informatics in Medicine Unlocked*, vol. 33, p. 101077, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352914822002131

[16] S. Das, M. S. Imtiaz, N. H. Neom, N. Siddique, and H. Wang, "A hybrid approach for bangla sign language recognition using deep transfer learning model with random forest classifier," *Expert Systems with Applications*, vol. 213, p. 118914, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422019327

[17] R. A. Nihal, S. Rahman, N. M. Broti, and S. Ahmed Deowan, "Bangla sign alphabet recognition with zero-shot and transfer learning," *Pattern Recognition Letters*, vol. 150, pp. 84–93, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865521002269

[18] K. K. Podder, M. E. H. Chowdhury, A. Tahir, Z. B. Mahbub, A. Khandakar, M. S. Hossain, and M. A. Kadir, "Bangla sign language (bdsl) alphabets and numerals classification using a deep learning model," *Sensors*, vol. 574, no. 2, Jan 2022. [Online]. Available: https://doi.org/10.3390/s22020574

[19] M. A. Rahaman, K. U. Oyshe, P. K. Chowdhury, T. Debnath, A. Rahman, and M. S. I. Khan, "Computer vision-based six layered convneural network to recognize sign language for both numeral and alphabet signs," *Biomimetic Intelligence and Robotics*, vol. 4, no. 1, p. 100141, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667379723000554

[20] M. S. Islam, S. S. S. Mousumi, N. A. Jessan, A. S. A. Rabby, and S. A. Hossain, "Ishara-lipi: The first complete multipurposeopen access dataset of isolated characters for bangla sign language," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*. IEEE, 2018, pp. 1–4.

[21] A. L. C. Ottoni, R. M. de Amorim, M. S. Novo, and D. B. Costa, "Tuning of data augmentation hyperparameters in deep learning to building construction image classification with small datasets," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 1, pp. 171–186, 2023.

[22] S. Azizi, S. Kornblith, C. Saharia, M. Norouzi, and D. J. Fleet, "Synthetic data from diffusion models improves imagenet classification," *arXiv preprint arXiv:2304.08466*, 2023.

[23] S. Faizal, C. A. Rajput, R. Tripathi, B. Verma, M. R. Prusty, and S. S. Korade, "Automated cataract disease detection on anterior segment eye images using adaptive thresholding and fine tuned inception-v3 model," *Biomedical Signal Processing and Control*, vol. 82, p. 104550, 2023.

[24] A. Mehmood, Y. Gulzar, Q. M. Ilyas, A. Jabbari, M. Ahmad, and S. Iqbal, "Sbxception: a shallower and broader xception architecture for efficient classification of skin lesions," *Cancers*, vol. 15, no. 14, p. 3604, 2023.

[25] B. Gülmez, "A novel deep neural network model based xception and genetic algorithm for detection of covid-19 from x-ray images," *Annals of Operations Research*, vol. 328, no. 1, pp. 617–641, 2023.