2024 3rd International Conference for Advancement in Technology (ICONAT)
Goa, India. Sep 13-14, 2024

Real-time Sign Language to Text Conversion using Deep Learning Models

Shriyan Shekhar
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong
sshekharaa@connect.ust.hk

2024 3rd International Conference for Advancement in Technology (ICONAT) | 979-8-3503-5417-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICONAT61936.2024.10774717

***Abstract*—For the communities of the deaf and hard of hearing, sign language is essential. In order to turn finger writing gestures into text, this work investigates deep learning models and MediaPipe-based systems for real-time recognition of American Sign Language (ASL). A range of methods are assessed, such as tailored CNN models, optimized architectures, and pre-trained models made with TensorFlow and Keras. The comparison demonstrates the effects of various model accuracies, training times, and designs. With validation accuracy of 95.6% and test accuracy of 97.8%, MobileNet shown potential for real-time ASL gesture classification. With a validation accuracy of 98.5% and a test accuracy of 99.5%, the CNN with MediaPipe fared better than the others, making it the best option for precise ASL gesture detection and text conversion. These results open the door for a real-time sign language to text converter by demonstrating how technology may improve sign language accessibility and comprehension.**

***Index Terms*—Sign Language, MediaPipe, ASL, CNN**

I. Introduction

The domains of computer vision and machine learning have made significant strides in the last several years, which have completely changed the way we work with visual data. The deaf and hard-of-hearing community benefits greatly from these technologies' potential in the field of sign language interpretation and recognition. The ability to communicate effectively between those who use spoken language and those who use sign language is made possible by sign language.

Students who are deaf or mute have several difficulties in conventional learning environments. The main barrier is communication because most professors and students do not understand sign language, which they frequently use. Their ability to fully engage in class activities and discussions may be hampered by this communication barrier, which can cause feelings of frustration and loneliness. To make matters worse, a lot of educational tools and materials are not created with accessibility in mind, which makes learning much more difficult.

34 million children and over 430 million other people worldwide—or more than 5% of the total population—need rehabilitation due to severe hearing loss. Over 700 million individuals, or one in ten, are expected to suffer a debilitating hearing loss by 2050. A loss of hearing in the better-hearing ear of more than 35 decibels (dB) is considered this form of hearing loss. Eighty percent of people with hearing loss who are unable to hear normally live in low- and middle-income nations. Age is a risk factor for hearing loss, with over 25% of adults over 60 having a deafening hearing loss [1]. Moreover, a lot of people who have hearing loss also have trouble speaking, so they need to use alternate forms of communication like sign language. The paper aims to offer technical solutions that can greatly enhance these people's communication experiences and increase their capacity to interact with the larger community. The main focus of this work is on the recognition of American Sign Language (ASL) finger-spelling alphabets, as seen in Fig 1 [2]. The objective is to create precise and effective models capable of real-time text translation using ASL finger spelling movements. This project seeks to empower people who use sign language as their primary form of communication, advance inclusion, and improve communication accessibility.

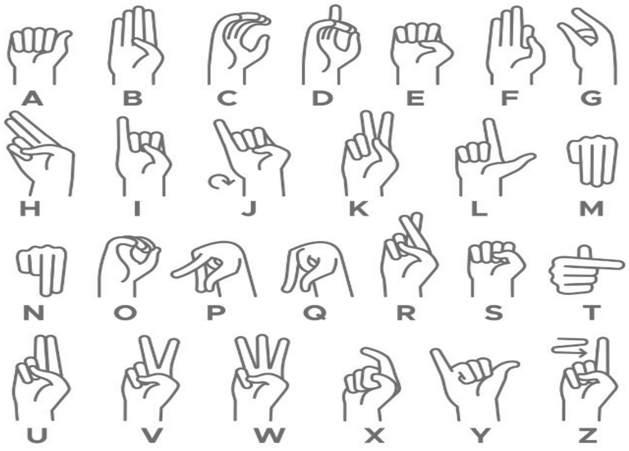

Fig. 1. American Sign Language Fingerspelling

This research focuses on ASL finger spelling recognition, which has less constraints than general sign language recognition since it only uses a small set of hand forms. Unlike several other sign languages, finger writing in ASL is done with a single hand, which minimizes hand occlusion difficulties and makes it a good place to start when creating recognition models. In order to accomplish the aim of real-time sign language to text translation, this research thoroughly examines a number of deep learning models, tools, and techniques.

It makes use of bespoke architectures, fine-tuning, and pretrained models with frameworks like TensorFlow and Keras. Furthermore, for reliable feature extraction from visual input and hand movement tracking for exact gesture identification, computer vision libraries like MediaPipe [3] and OpenCV must be included.

The field of educational technology, or EdTech, is poised to gain greatly from developments in sign language recognition. By integrating these technology into classrooms, educators may create inclusive learning environments where students who are hard of hearing or deaf can interact with peers and teachers more successfully. Teachers may create tools that enable real-time translation of sign language into text by utilizing deep learning and computer vision. This will help to close communication gaps and improve the educational experience for students who use sign language.

This paper's following sections will go into detail on the research's techniques, experimental designs, findings, and implications. This thorough investigation will clarify the nuances of sign language interpretation and demonstrate how technology may revolutionize the field of education technology by fostering inclusive communication. Through the creation of effective and precise models for ASL finger spelling recognition, this research seeks to significantly advance the accessibility and inclusivity of future developments for people who use sign language.

II. Related Works

In order to close communication barriers for the deaf and hard of hearing people, sign language recognition has seen a number of innovative approaches utilizing computer vision and machine learning.

According to Fernandes et al.'s paper [4], "Convolutional Neural Network based Bidirectional Sign Language Translation System," CNNs were used to create a bidirectional sign language translation system. They started off using a hardware glove with flex sensors, but eventually switched to a software-based method where CNNs trained on a variety of datasets greatly increased accuracy. According to Fernandes et al., their system exhibits great adaptability and customisation possibilities since it can translate sign language to text and voice and vice versa.

In "Sign Language to Text Conversion in Real Time using Transfer Learning," Thakar et al. [5] presented a deep learning model that uses the VGG16 architecture in conjunction with CNN to convert ASL finger spelling to text in real time. Their method increased accuracy using transfer learning from 94% with CNN to 98.7%. In order to provide a reliable ASL translation solution, they created an application that uses a camera to record user motions, interprets the pictures using a Django backend, and then predicts the associated text.

In "Conversion of Sign Language into Text," Mahesh Kumar N B [6] used MATLAB to investigate sign language recognition for Indian Sign Language (ISL). Preprocessing, hand segmentation, Eigen values and vectors feature extraction, and Linear Discriminant Analysis (LDA) gesture detection are all included in the system. Gestures that are recognized are translated into voice and text forms. According to Kumar, this technique showed promise for lowering the dimensionality and noise in the data, which would increase identification accuracy.

In their work "Sign Language to Text Conversion using Hand Gesture Recognition," Tanya Kemkar [7] and colleagues concentrated on CNN-based ASL recognition. Their technique entails filtering photos beforehand and then classifying them using a CNN. It achieves 99% accuracy on the MNIST dataset and 96% accuracy on the ASL dataset. According to Kemkar et al., the CNN-based method shows good recognition accuracy when it comes to ASL finger writing movements, proving to be an efficient solution to computer vision problems.

In "Sign Language Conversion to Text and Speech," Medhini Prabhakar et al. [8] created a system for identifying ISL hand gestures that made use of a number of models, including CNN, FRCNN, YOLO, and MediaPipe. Using a camera, the system records hand motions, applies several algorithms to the photos, and then translates the identified gestures into text and voice. According to Prabhakar et al., the MediaPipe model demonstrated superior accuracy and real-time conversion performance, rendering it appropriate for use in real-world scenarios.

In their paper "Breaking the Silence: An innovative ASL to Text Conversion System Leveraging Computer Vision & Machine Learning for Enhanced Communication," Bagane et al. [9] developed an inventive ASL to text conversion system. The technology recognizes ASL motions reliably and converts them into text quickly using machine learning and computer vision techniques. The goal of this system is to improve communication for the deaf and hard of hearing people. It consists of a letter recognition model, a gesture recognition module, and a text production module. Their method emphasizes the possibility of smooth communication in a number of contexts, such as public contacts, healthcare, and education.

In "Conversion of Sign Language to Text and Speech and Prediction of Gesture," Bharath A Manoj [10] introduced a device utilizing an Arduino Uno board, flex sensors, and an Android application to convert sign language to text and speech. The system detects hand gestures via flex sensors, sending the data to an Android device through a GSM module, which translates the text message into speech. This device also predicts gestures by sending sensor inputs to a cloud-based server for future reference, aiming to continuously improve reliability by learning user behaviors.

Together, these studies demonstrate the progress made in sign language recognition via the use of both software and hardware. Software solutions utilizing deep learning frameworks such as CNNs and transfer learning have demonstrated significant gains in accuracy and versatility, while hardware-based approaches provide physical engagement. The efficiency and scalability of sign language recognition systems are continually improved by the integration of computer vision libraries and cutting-edge neural network designs.

III. Methods

Finding an appropriate dataset for deep learning proved to be difficult at first, which prompted the idea of making a bespoke dataset. Nevertheless, the emphasis moved to finding publicly accessible datasets because of time constraints and different legal, social, and ethical considerations. The Kaggle ASL alphabet dataset turned out to be a great option [11]. This dataset has garnered significant attention, with about 267 notebooks and 11 talks on Kaggle. It is widely utilized by researchers and students for machine learning and sign language recognition applications.

There are 87,000 200x200 pixel images in the dataset, distributed among 29 classifications. These are three more classes called Delete, Space, and Nothing in addition to the 26 classes for the ASL alphabet. A total of 27 classes—26 letter classes and the Space class—were chosen for this study, which attempts to transform sign language to text. With 3,000 photos in each class, the total number of photographs in the collection is 81,000.

A. Dataset

In order to identify the best deep learning model for real-time applications, this study compares a number of pre-trained models, including both customized and fine-tuned models. Eighty percent of the dataset was set aside for training, while the remaining twenty percent was set aside for validation. There was just one test image per class in the initial Kaggle ASL dataset. The class names were changed from alphabetical to numerical labels in order to expedite the categorization of images. More samples were created and enhanced the data to create a new test dataset, making sure that every class has 100 photos for testing. While fine-tuning was done on 128x128 photos, pre-trained models were trained on RGB images that had been scaled to 224x224 pixels. Additionally, custom CNN and ResNet models were evaluated using Gaussian blur, grayscale, and binary images for classification.

B. Image Processing

The dataset was improved by applying several transformations to the original photos using TensorFlow/Keras' ImageDataGenerator, which improved the model's capacity to generalize and function effectively on fresh data. Pixel values were normalized using the rescale option, which divided them by 255 to guarantee that they were inside the [0, 1] range. Random rotations up to 15 degrees were possible with the rotation range argument. Shifts in both the horizontal and vertical directions up to 10% of the picture dimensions were made possible by the width and height shift range parameters. Shear range controllable shear transformations produced x- or y-axis slanted pictures. The horizontal flip function made it possible to flip pictures horizontally, which helped the model identify characteristics from various orientations. The zoom range parameter allowed for up to 20% in and out zooming.

C. Technologies Used

The ASL alphabet dataset was trained utilizing cutting-edge technology, namely TensorFlow and Keras to create complex neural networks. Preprocessing was made possible via OpenCV, which made it possible to extract features from visual data efficiently. It managed intricate vision tasks, whereas PIL provided low-power image processing features. Hand landmarks were extracted using MediaPipe, which is useful for deciphering sign language. Different assessment criteria were supplied by TensorFlow/Keras and SciKit-Learn. SpellChecker was utilized to improve the correctness of ASL to text conversion by recommending fixes, and Tkinter was used to construct the real-time conversion interface.

D. Learning Techniques Used

1) Supervised Learning - Models are trained using labeled data in supervised learning, a basic machine learning technique. This method uses a collection of sign images linked with matching text labels (A-Z) to translate ASL alphabet signs into text. The basis for training the model is formed by the representation of each character in each image. The photos in the ASL dataset are labeled with the corresponding letters, dividing it into training and validation sets. These photos are used to train deep learning models, such as CNNs and RNNs, to predict text labels. Using loss functions to gauge prediction accuracy, the model learns the associations between the labels and pictures during training. The weights of the model are then modified by optimization methods to reduce the loss. The model's efficacy is assessed on the validation set to ensure proper ASL to text conversion.
2) Transfer Learning - Transfer learning makes use of information from one model to enhance the process of learning from another. This entails employing pre-trained algorithms learned on huge datasets such as ImageNet for ASL alphabet sign language translation. The first step in the procedure is to choose a base model that has already been trained, such as VGG, ResNet, or MobileNet. This model's layers recognize important patterns and structures in pictures by acting as feature extractors. Subsequent layers designed for the conversion of ASL alphabets are fed these attributes. The ASL alphabet dataset is used to fine-tune the model's weights, concentrating on certain layers to better capture the subtleties of ASL signals. Using the ASL dataset, the model is trained and assessed, utilizing the pre-trained model's prior knowledge to accelerate convergence and improve generalization.

E. Methodology for Pre-Trained Models

The aim is to evaluate pre-trained models for text-to-sign language conversion in terms of training time and accuracy. TensorFlow, Pandas, Matplotlib, and Seaborn are used by the code to handle, train, visualize, and evaluate data. By minimizing wasteful training and maximizing resource use, this comparison framework aids in decision-making. The best solutions are found through the analysis of model strengths, which improves generalization and development. The evaluation procedure is metric-based and transparent, which facilitates stakeholder communication and decision-making. The adaptability of model comparison enables customized decisions based on the requirements of the project, including complexity, interpretability, and time restrictions.

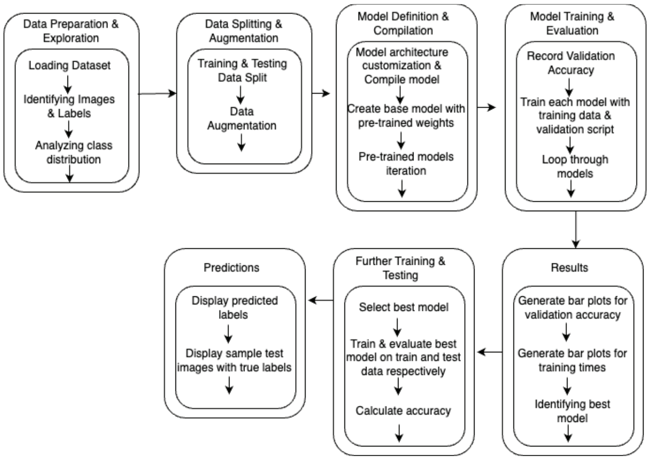
Data Preparation & Exploration
Loading Dataset
Identifying Images & Labels
Analyzing class distribution
Data Splitting & Augmentation
Training & Testing Data Split
Data Augmentation
Model Definition & Compilation
Model architecture customization & Compile model
Create base model with pre-trained weights
Pre-trained models iteration
Model Training & Evaluation
Record Validation Accuracy
Train each model with training data & validation script
Loop through models
Results
Generate bar plots for validation accuracy
Generate bar plots for training times
Identifying best model
Further Training & Testing
Select best model
Train & evaluate best model on train and test data respectively
Calculate accuracy
Predictions
Display predicted labels
Display sample test images with true labels

Fig. 2. Pre-trained Models Methodology

The process starts with loading and categorizing the ASL alphabet dataset as shown in Fig. 2., which has pictures of the letters used in sign language. Through the analysis of the class distribution and picture visualization, the dataset is studied. TensorFlow's ImageDataGenerator uses data augmentation techniques to provide variable training data, which improves model generalization, after separating the dataset into training and testing subsets. Pre-trained architectures such as VGG, ResNet, MobileNet, EfficientNet, DenseNet, InceptionV3, and Xception are customized with dense layers for the ASL conversion problem in transfer learning.

Accuracy measures, the Adam optimizer, and the categorical cross-entropy loss function are used to construct pre-trained models, which are given in dictionary format. Following that, the models are monitored using validation data and trained using training data. For every model, training durations and validation accuracy data are documented. Bar charts make training timeframes and validation accuracy visible, which helps choose the best model based on performance. With the combined training and validation data, the selected model is subjected to additional training. Metrics like accuracy are used to assess performance on the test data, and the results are displayed using a normalized confusion matrix that displays classification performance across classes. A clear representation of the model's predictions and performance insights is given by the sample test photos that have both true and forecasted labels.

F. Custom CNN Models

Convolutional Neural Networks (CNNs) are specialized neural networks that are vital for computer vision jobs since they are made to analyze grid-like input, such pictures and movies. They incorporate many essential elements and automate feature extraction like Convolutional Layers for Texture and Edge Recognition, Layer pooling to reduce spatial dimensionality, Completely Networked Layers for ultimate classification, Initiation serves to bring about non-linearity, Data are flattened to create one-dimensional vectors, Dropout to avoid overfitting, Softmax-activated output layer for ultimate predictions. In order to optimize and evaluate performance, the model is assembled with an optimizer, loss function, and assessment measures prior to training.

1) Building CNN Architecture: The Sequential API is used in the construction of the CNN architecture enabling the insertion of layers one at a time. There are twenty training epochs. The architecture as shown in Fig. 3. starts with max-pooling with a 2x2 pool size, then moves on to a convolutional layer with 32 filters and a 3x3 kernel utilizing ReLU activation. The same parameters apply to the next convolutional layer, which is also accompanied by max-pooling. After that, the output is compressed into a one-dimensional vector. Next come fully linked (Dense) layers with 128, 96, and 64 units. To avoid overfitting, each of these layers has dropout layers with a 40% dropout rate and uses ReLU activation. For the ASL alphabet classes, the final dense layer consists of 27 units with softmax activation for classification.

The model measures the difference between the actual and predicted classes using categorical cross-entropy loss, and it assesses the accuracy of the predictions using accuracy as a metric. Zooming, Shearing, and flipping are examples of data augmentation techniques that are used to improve the dataset's variety and offer reliable performance.

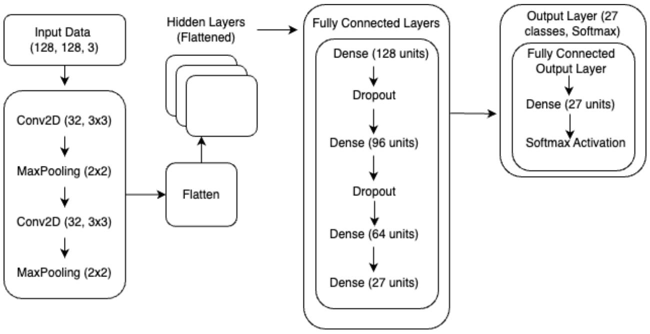

Fig. 3. CNN Architecture

G. CNN with MediaPipe

Google's open-source MediaPipe infrastructure is essential for translating ASL to text. It enables real-time interpretation and synchronization with user operations by precisely tracking hand movements. Its flexible algorithms interpret ASL movements accurately, and its pre-built models make app creation easier and increase accuracy.

1) Building CNN-MediaPipe Architecture:: Using the MediaPipe hands module, the architecture first processes RGB pictures in order to extract hand landmark coordinates. Lists containing these landmarks and their matching labels are used to train a CNN model to recognize sign language. The CNN uses the retrieved X and Y coordinates from landmark detections as input features since they represent hand points as two-dimensional pixel values.

For feature extraction and pattern identification from the hand landmarks, the architecture incorporates Conv1D and MaxPooling1D layers. Conv1D (32 filters, kernel size 3, ReLU activation) and MaxPooling1D (pool size 2) are the first two layers in the sequence. Repeating this process results in a MaxPooling1D layer (pool size 2) and another Conv1D layer (64 filters, kernel size 3, ReLU activation) that captures detailed features and preserves information through max pooling.

To create a one-dimensional vector from the output of the Conv1D and MaxPooling1D layers, the completely linked levels begin with a Flatten layer. A Dropout layer (0.5 rate) and a Dense layer with 128 units (ReLU activation) come next to avoid overfitting. Classification based on hand landmarks is made possible by the last Dense layer, which has 27 units (softmax activation) to calculate the probability for the 27 ASL alphabet letters.

Conv1D and MaxPooling1D layers are interleaved to help with dimensionality reduction and feature extraction. The last dense layer with softmax activation computes the probability for ASL class predictions, while the flatten layer prepares the input for fully connected layers. The accuracy of real-time ASL gesture interpretation is finally improved by Conv1D layers learning features and pooling layers retaining important information and reducing dimensions.

H. Fine-Tuned Models Methodology

Three effective CNN architectures that are tailored for image categorization on different platforms are MobileNet, VGG, and EfficientNet. Custom layers are added, creating a new architecture, to customize these pre-trained models for ASL alphabet categorization. With inputs set to the original dimensions (128x128, RGB), and outputs producing predictions from the modified layers, the Model class assembles the whole structure.

A global average pooling layer is added to the pre-trained MobileNet model for ASL alphabet classification in order to minimize the feature map dimensions. This is followed by a dense layer with 1024 units and ReLU activation in order to aggregate features. Class probabilities are generated using softmax activation in the last dense layer. The Adam optimizer and categorical cross-entropy loss are utilized in the compilation of this unique model. The fit technique with train generator and validation generator for data is used in the training process. EarlyStopping is used to stop training if the validation loss does not improve after a predetermined number of epochs.

I. Other Custom Models

Additionally, Experimentation was done with Vision Transformer (ViT) and a Custom ResNet. After three training attempts, Investigating ViT was not possible any further due to GPU restrictions. ResNet fared the best among these models when it came to binary pictures.

In very deep networks, the vanishing gradient problem is addressed with the ResNet design. Batch normalization and ReLU activation come after a convolutional layer (64 filters, 7x7 kernel, 2x2 stride) on the input. Dimensions are decreased using a max-pooling layer (3x3 pool size, 2x2 stride). Deep network training is improved by residual blocks, which are composed of two 3x3 convolutional layers, batch normalization, ReLU activation, and skip connections. A feature vector created via global average pooling condenses spatial input into units that correspond to the number of classes (for example, 27 for ASL letters) and softmax activation for class probabilities. The feature vector is then connected to a dense layer. With deep layers computing probability and pooling condensed data, this architecture collects features in a hierarchical manner. To avoid overfitting, the model makes use of the Adam optimizer, categorical cross-entropy loss, and early stopping. Functions from Keras help with test assessment, training with generators, visualization, and model summary creation. For future use, the architecture and weights are stored.

J. Evaluation Methodologies

Different assessment techniques shed light on how well the model recognizes ASL gestures.

1) Accuracy score: The accuracy score measures how frequently the model is right by dividing the number of correct predictions by the total number of occurrences. It may not be enough, nevertheless, in cases with unbalanced data or different implications of misclassification.
2) Classification report: The classification report offers a comprehensive overview of the model's performance for every class. It includes information on precision (the percentage of correctly predicted positives among all predicted positives), recall (the percentage of correctly predicted positives among all actual positives), support (the number of instances per class), and the F1-score (the harmonic mean of precision and recall).
3) Test Accuracy: This gauges the model's performance on hypothetical test data, evaluating how well it generalizes to novel and unfamiliar situations and providing insights into actual performance.
4) Validation Accuracy: This measures the model's performance on a different validation set during training. If validation accuracy starts to drop, it may indicate that the model is overfitting and that there is a discrepancy with fresh data.
5) Confusion Matrix: This shows mistakes like false positives/negatives and true positives/negatives by tabulating projected against real class occurrences. Strong model performance and accurate predictions are shown by high values along the diagonal.

K. Methodology of Real-Time Application

Using the weights from the chosen model, a real-time application was created for translating ASL alphabet signs to text after determining which model had the best validation accuracy. Tkinter was used in the development of the application's graphical user interface (GUI). In addition to offering an entry area for the current word and a "Clear Last Character" button with word completion suggestions, it predicts characters and phrases.

The program continually records frames from a camera by using the update frame method. The hand tracking model uses these frames to forecast motions. Predicted characters, potential words, and recommendations for finishing words are updated in real-time in the GUI. By pointing at places and making gestures using the ASL alphabet, users may spell words aloud.

IV. Results and Discussion

Using important measures including accuracy, confusion matrix, F1 score, precision, and recall, we assessed the models' performance in this section. These metrics provide a thorough evaluation of the models' recognition accuracy for ASL letter movements. Each model's advantages and disadvantages were emphasized in the research, enabling meaningful comparisons and useful findings. The model with the highest performance was chosen to create a Python application that converts ASL sign language to text in real-time.

1) Pre-Trained Models Results: These models were evaluated primarily on the basis of validation accuracy, training duration, and overall performance. A CSV file containing the validation times and accuracy was created, and the data was sorted by validation accuracy.

With a validation accuracy of 95.6%, MobileNet fared better than the other 27 pre-trained models, according to the results. This model was kept for later analysis. MobileNet produced a normalized confusion matrix, computed accuracy, predicted labels on the test set, and showed example pictures with true and predicted labels. MobileNet achieved a test accuracy of 97.8%, indicating outstanding performance.

After a comprehensive review, it was determined that MobileNet was the best pre-trained model for classifying ASL letter gestures, correctly predicting test pictures with both real and anticipated labels.

TABLE I
Validation Accuracy

Model	Accuracy
MobileNet	95.6
DenseNet201	93.1
MobileNetV2	92.5
ResNet152V2	91.2

2) Custom CNN Model Evaluation: The validation accuracy of the customized CNN model was 97.5%. A test accuracy and accuracy score of 98.4% was obtained after additional analysis on a dataset of 3500 test pictures comprising accuracy score, test accuracy, classification report, and confusion matrix.

The majority of the diagonal members in the confusion matrix had high values, suggesting that many classes had correct predictions and that there were not many misclassifications. The majority of the classes in the categorization report had good accuracy, recall, and F1 scores; however, certain classes, including 20 and 21, had significantly lower scores, indicating areas that needed work. On the test data, the model did well overall.

3) CNN with MediaPipe Evaluation: The validation accuracy for the CNN with MediaPipe model was 98.5%. A total of 100 test samples per class were used in the assessment procedure for the ASL gesture classification job, yielding an accuracy score of 99.5% as well as an accuracy matrix and classification report. Fig 4. depicts the Validation Accuracy and Fig. 5. depicts the Validation Loss.

The classification report demonstrated good competence in correctly identifying ASL letter motions, as evidenced by the high accuracy, recall, and F1 scores. With high diagonal values and comparatively low off-diagonal values, the confusion matrix demonstrated a good match between the model's predictions and the real classes, indicating accurate performance.

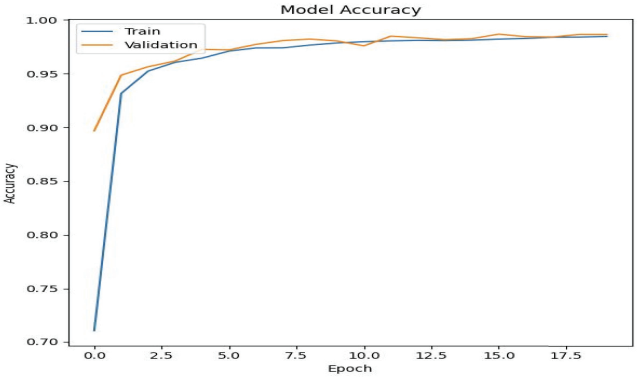

Fig. 4. Validation Accuracy

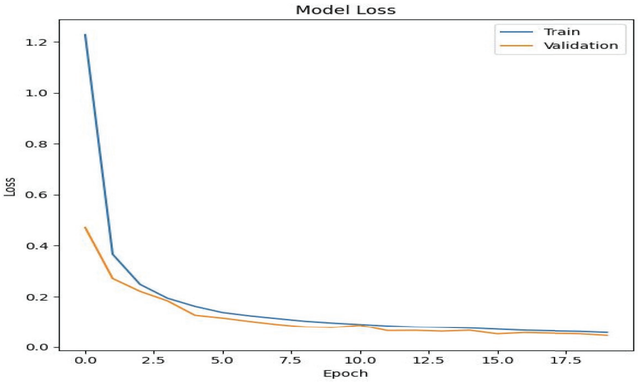

Fig. 5. Validation Loss

4) Fine-Tuned MobileNet Evaluation: When it came to pre-trained model validation accuracy, MobileNet performed quite well. After adjustments, its validation accuracy increased to 96.

Test accuracy and accuracy score of 98.6% were obtained through performance analysis of the test data; this is somewhat less than the test accuracy of the pre-trained MobileNet. Although certain classes differed, the classification report showed good overall performance with excellent recall, accuracy, and F1 scores. In addition to several misclassifications, such as class 4 (alphabet E) being projected as class 5 (alphabet F) and class 15 (alphabet P) being forecasted as class 14 (alphabet O), the confusion matrix had significant diagonal values.

5) Custom ResNet Evaluation: The customized ResNet model obtained a test accuracy and accuracy score of 91.1% along with a validation accuracy of 95.3%.

Although class 0 had 95% accuracy, the classification report showed that it only recorded 75% of real cases (recall). Class 1 demonstrated both accuracy and leniency in predictions, with a high precision and a recall. Overall, the model's advantages and disadvantages varied depending on the class. When individual and class-weighted outcomes were taken into account, the macro and weighted averages gave a comprehensive picture of its performance throughout all classes.

6) Finding the Best Model for Real-Time Application: The CSV file produced during the assessment of pre-trained models was updated with the validation accuracies of the custom CNN, custom ResNet, fine-tuned MobileNet, and CNN with MediaPipe. According to the highest validation accuracy, the CSV file was sorted.

Based on these findings, CNN using MediaPipe had the best accuracy in testing and validation. As a result, it was chosen to create the real-time ASL sign language to text application.

V. Conclusion

This study offers a thorough analysis and comparison of many deep learning models for text translation from American Sign Language (ASL) alphabet motions. The study assesses bespoke architectures, fine-tuning techniques, and pre-trained models, emphasizing the real-time applicability of each. After a careful examination, we determined that the CNN with MediaPipe model is the most effective choice for precise and effective ASL to text translation in real-time scenarios.

Our real-time ASL to text converter offers sign language users substantial advantages by precisely identifying ASL motions, creating words, and recommending spellings, so demonstrating the efficacy of our research. This study emphasizes the value of customized model selection, optimization, and cutting-edge methods like MediaPipe for enhancing the identification of ASL letter gestures. By overcoming communication barriers, these methods will improve the development of assistive technology and increase inclusivity.

Subsequent investigations in this domain may concentrate on enhancing the precision of the models, especially for motions that entail comparable hand movements. It could be beneficial to improve the application's accessibility and usefulness by adding user-friendly interfaces and adding more language support. In the final analysis, our work advances assistive technology and promotes smooth communication across various cultures.

References

[1] https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] https://images.app.goo.gl/DDXKqcm1bbb8mkHe6

[3] Lugaresi, Camillo Tang, Jiuqiang Nash, Hadon McClanahan, Chris Uboweja, Esha Hays, Michael Zhang, Fan Chang, Chuo-Ling Yong, Ming Lee, Juhyun Chang, Wan-Teh Hua, Wei Georg, Manfred Grundmann, Matthias. (2019). MediaPipe: A Framework for Building Perception Pipelines.

[4] L. Fernandes, P. Dalvi, A. Junnarkar and M. Bansode, ”Convolutional Neural Network based Bidirectional Sign Language Translation System,” 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2020, pp. 769-775, doi: 10.1109/ICSSIT48917.2020.9214272.

[5] S. Thakar, S. Shah, B. Shah and A. V. Nimkar, ”Sign Language to Text Conversion in Real Time using Transfer Learning,” 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2022, pp. 1-5, doi: 10.1109/GCAT55367.2022.9971953.

[6] Mahesh Kumar N B, ”Conversion of Sign Language into Text,” International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7154-7161

[7] T. Kemkar, V. Rai and B. Verma, ”Sign Language to Text Conversion using Hand Gesture Recognition,” 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 1580-1587, doi: 10.1109/ICCES57224.2023.10192820.

[8] Prabhakar, Medhini & Hundekar, Prasad & Deepthi, Sai & Tiwari, Shivam & Ms, Vinutha. (2022). SIGN LANGUAGE CONVERSION TO TEXT AND SPEECH. 9.

[9] Bagane, Pooja & Thawani, Muskaan & Singh, Prerna & Ahmad, Raasha & Mital, Rewaa & Amenu, Obsa. (2024). Breaking the Silence: An innovative ASL-to-Text Conversion System Leveraging Computer Vision and Machine Learning for Enhanced Communication. International Journal of Intelligent Systems and Applications in Engineering. 12. 246–255.

[10] A, Bharath. (2020). Conversion of Sign Language to Text and Speech and Prediction of Gesture. International Journal of Recent Technology and Engineering (IJRTE). 8. 4191-4194. 10.35940/ijrte.F9502.038620.

[11] ”ASL Alphabet.” Kaggle, www.kaggle.com/grassknoted/asl-alphabet.

[12] S. B. Reddy, A. Shahebaaz, R. Dodda and C. Raghavendra, ”Transformative Advancements: Sign Language Conversion to Text and Speech,” 2024 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2024, pp. 1-7, doi: 10.1109/ESCI59607.2024.10497370.

[13] A. Maitrey, V. Tyagi, K. Singhal and S. Gupta, ”A Framework for Sign Language to Speech Conversion Using Hand Gesture Recognition Method,” 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2023, pp. 333-338, doi: 10.1109/CICTN57981.2023.10140730.

[14] S. R, S. R. Hegde, C. K, A. Priyesh, A. S. Manjunath and B. N. Arunakumari, ”Indian Sign Language to Speech Conversion Using Convolutional Neural Network,” 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-5, doi: 10.1109/MysuruCon55714.2022.9972574.

[15] M. S. Nair, A. P. Nimitha and S. M. Idicula, ”Conversion of Malayalam text to Indian sign language using synthetic animation,” 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), Kottayam, India, 2016, pp. 1-4, doi: 10.1109/ICNGIS.2016.7854002.

[16] M. Sultana, J. Thomas, S. Thomas, M. SA and S. L. S, ”Design and Development of Teaching and Learning Tool Using Sign Language Translator to Enhance the Learning Skills for Students With Hearing and Verbal Impairment,” 2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE), Vellore, India, 2024, pp. 1-5, doi: 10.1109/ic-ETITE58242.2024.10493342.

[17] N. Addepalli, R. K. Pabolu, S. Ganesh Chennuru, V. L. Vissampalli and G. L. Madhumati, ”Conversion of American Sign Language to Text Using Deep Learning for Feature Extraction and Naive Bayes for Classification,” 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-5, doi: 10.1109/I2CT57861.2023.10126201.

[18] Suharjito, G.H., Thiracitta, N., Nugroho, A. (2018). Sign language recognition using modified convolutional neural network model. 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), Jakarta, Indonesia. https://doi.org/10.1109/INAPR.2018.8627014