SIGN LANGUAGE RECOGNITION USING DL

J V badrinath[1], M Dineswar[2], K Mahesh Reddy[3], K Damarukesh Reddy[4],
Ms. R Sathiya Priya[5], Dr. R Karunia Krishnapriya[6],
[1,2,3,4] UG Scholar, [5]Assistant Professor, [6]Associate Professor, Department of CSE,
Sreenivasa Institute of Technology and Management Studies, Chittoor, India

Abstraction : For the deaf and hard-of-hearing communities, sign language is an essential form of communication. In this study, we use skeleton-based hand gesture photos to demonstrate a deep learning-based method for real-time sign language detection. The skeleton structure of hand movements was captured as white lines on black background using camera input and Mediapipe to create a customized dataset of alphabetic signs(A-Z). Up to 200 resized 224x224 pixel photos were used to represent each class. Using this dataset, MobileNetV2, a lightweight convolutional neural network designed for embedded and mobile vision applications, was refined to carry out multi-class classification. The trained model demonstrates high accuracy in classifying isolated static signs and supports interactive prediction through a webcam interface.

Keywords: Deep Learning, Sign Language Recognition, MobileNetV2, Skeleton-based Images, MediaPipe, Human-Computer Interaction.

I. INTRODUCTION

For many people with hearing loss, sign language is their main form of communication, education, an accessibility tools can all be substantially improved by automatically recognizing sign motions. Conventional vision-based techniques frequently use RGB photos or sensor data, which are impacted by lighting, background noise, and reliable substitute for gesture recognition is skeleton-based representations.

Using a MobileNetV2 deep learning model and a proprietary dataset of skeleton-based photos, this work attempts to create an end-to-end real time identification system that can identify static sign gestures (A-Z). Using MediaPipe and webcam input, a custom dataset was created that captured the hand landmarks for each of the 26 letters (A-Z) in American Sign Language (ASL) as white skeleton lines on a black background. Up to 200 photos were added to each class, and the data was preprocessed and scaled to 224x224 pixels.

The system also offers a new option based interface in addition to its basic prediction capabilities. When a trigger key is pushed, the top three model predictions are shown on the screen. By removing inaccurate recommendations, users enable the system to only commit to the final, verified letter. Even we form a word by choosing the letter by letter to form a complete word. This allows for partial human supervision of the prediction flow, improves user control, and lowers error rates.

II. LITERATURE SURVEY

Sensor gloves and manual feature extraction were used in early sign language recognition techniques. By automating feature learning, vision-based deep learning models have made great progress in the field. Using RGB pictures, CNNs like AlexNet and VGG have been utilized to classify gestures. But because skeleton-based techniques are not affected by background, color, or lighting, they are becoming more and more popular.

Real-time hand skeleton extraction from webcam input is now possible because to Google's MediaPipe, and this may be used to create a consistent gesture collection. Dynamic gesture recognition has showed potential in studies on skeleton sequences utilizing models like as 3D CNNs and GCNs. MobileNetV2 offers a competitively accurate and highly efficient architecture for static gestures, making it ideal for embedded and mobile devices.

Several researchers have investigated depth and infrared imaging for sign language Identification in addition to RGB-based and skeleton-based approaches. Although these modalities provide better hand posture estimation and spatial information, their accessibility for regular users is limited since they require specialized hardware like Kinect or Intel RealSense. As such, their use is still limited to regulated settings.

Transformer-based models and temporal learning for sequential gesture interpretation arre the subjects of another area of study. Models such as Vision Transformers(VIT) and Spatio-Temporal Graph Convolutional Networks(ST-GCN), which are mainly used in dynamic sign recognition, have demonstrated notable advancements in learning temporal dependencies across hand movement, which may be advantageous when switching from static to continuous sign recognition.

III. METHODOLOGIES

This study uses skeleton-based hand images to provide a deep learning-based framework for static sign language gesture identification. Real-time hand landmark extraction is used for data collection, image preprocessing, transfer learning for model training with MobileNetV2, hyperparameter tuning, model evaluation, interpretability analysis, and real-time prediction via an interactive user interface are all included in the methodology. To confirm the suggested system's viability in practice, a working prototype was created.

Data Collection

The MediaPipe framework, which enables precise hand landmark identification, was used in conjunction with a webcam to acquire the dataset in real-time. At a resolution of 224 x 224 pixels, the extracted landmarks were shown as white skeletal structures on a consistent black backdrop. We gathered up to 200 photos for every letter (A-Z) in the alphabet. During model training and validation, these were kept in class-specific directories to provide effective organization and accessibility.

Preprocessing

the dataset was standardized by resizing each image of 224 by 224 pixels and normalizing them to a range of 0 to 1. To emulate real-world variability in hand pose presentation and improve model robustness, simple data augmentation techniques were used, such as small rotations, scaling, and translations. The dataset, which made up 80% and 20% of the total data, respectively, was divided into training and validation subsets. IN addition to enabling performance tracking and generalization evaluation, this division guaranteed efficient model learning.

Algorithms and Learning Models

The MobileNetV2 convolutional neural network, which was chosen for its lightweight design and excellent performance on embedded and mobile devices, servers as the foundation for suggested categorization system. The model was adjusted for the ASL recognition challenge after being pre-trained on the built dataset. A global average pooling layer, a dense layer with 256 neurons driven by ReLU, a dropout layer to reduce overfitting, and a SoftMax output layer set up for 26 class classification in accordance with ASL alphabet were used in place of the original classification layers

Hyperparameter Tuning

To get the best model performance, extensive hyperparameter changing was done. A number of parameters were rigorously assessed, including training epochs (varying from 20 to 50),dropout rate (0.3,0.4,0.5),learning rate(0.001,0.0005,0.0001), and batch size(16,32,64). The optimal configuration was identified as a batch size of 32.

Model Evaluation

Several assessment criteria, such as overall accuracy, precision, recall, and F1-score for each class, were used to objectively evaluate the models performance. Class-wise prediction accuracy was also examined using a confusion matrix. Strong predictive power and generalization across all gesture classes are demonstrated by the models consistent validation accuracy above 95%. Even though there were sporadic misclassification between signs that looked alike, performance was consistent throughout the sample.

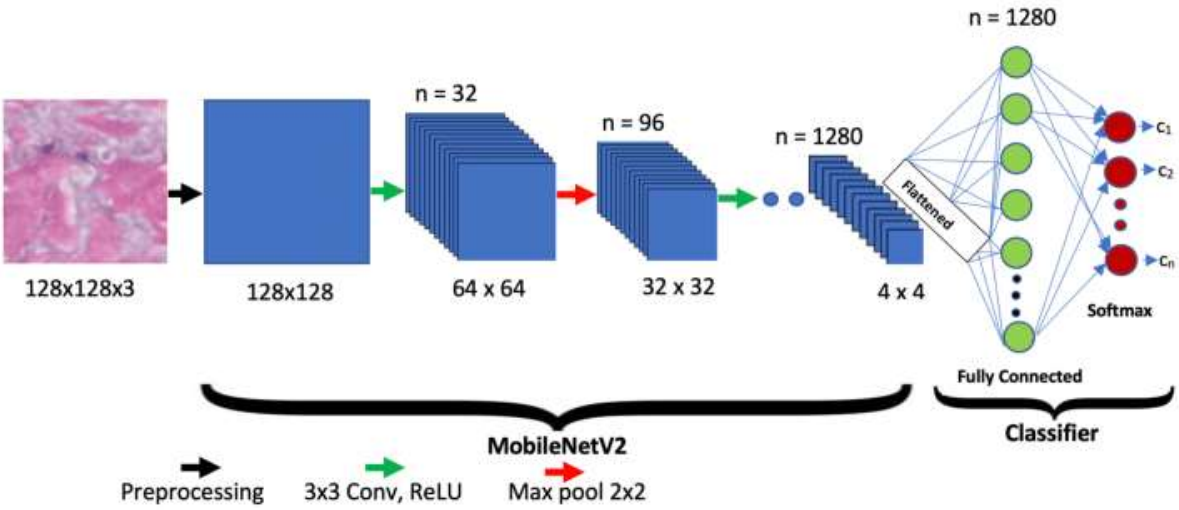

Fig.1 Architecture Diagram of MobileNetV2

Model Interpretability

Class activation maps (CAMs) were used to depict the hand regions that made the biggest contributions to model predictions in order to improve interpretability and transparency. The prototype also included a top- three prediction technique, which lets users choose he best forecast from the outputs with the highest likelihood.

Real-Time Prediction System

Dynamic gesture recognition using webcam input was made possible by the implementation of a real-time prediction engine. A frame is taken when a specific key is pressed, and MediaPipe is used to extract hand landmarks from the frame. These and marks are then fed into the trained model. The user can verify the intended gesture by the looking at the top three predict labels that the system displays. The chosen character is added to a forming word and shown on the screen when it has been verified. Prediction mistakes are successfully decreased and the user experience is improved.

Prototype Development

A completely functional prototype was created using Python to show the suggested frameworks's practical usefulness. The system incorporates TensorFlow/Keras for deep learning interference, MediaPipe for hand tracking, and OpenCV for the user interface and real-time video recording. In addition to supporting possible deployment on mobile or embedded platforms, the lightweights architecture guarantees effective performance on common hardware. The prototype demonstrates the potential of the suggested approach for assistive communication technologies and verifies its efficacy in practical situations.

Algorithms Used

MobileNetV2

The lightweight convolutional neural network architecture known as MobileNetV2 was created to operate effectively on mobile and embedded devices. It maintains great accuracy while lowering computational complexity by using depth wise separable convolutions. Inverted residual blocks with linear bottlenecks are introduced in this architecture to help maintain representational power while using fewer parameters. 26 static sign language gestures (A-Z) were multi-class classified using the pre-trained MobileNetV2 model, which was refined for this study using the created dataset. The fundamental convolutional layers of the model were kept in order to take use of learned feature representations, while the final dense layers were adjusted to fit the classification objective. The model was selected because to its low latency and accuracy, which makes it perfect for real-time sign recognition jobs on system with limited resources.

MediaPipe

Google created the open-source MediaPipe framework, which offers effective real-time perception pipeline solutions, such as hand tracking. In order to create skeleton-based gesture graphics, 21 hand landmarks were extracted from camera input using MediaPipe. The deep learning model used these skeletal representations as its main input. The creation of a consistent, clean dataset without the need for intricate segmentation or background filtering procedures was made possible by Mediapipe's real-time performance and resilience to changing backgrounds and lighting conditions.
For static sign language recognition in real-time settings, the combination of MediaPipe for input preprocessing and MocileNetV2 for classification offered a practical and efficient solution.

IV. RESULTS AND DISCUSSIONS

Experimental Setup

Total Records: 5200

Dataset used: Skeleton Sign Language Dataset

Tools: Python

Hardware: Windows with CPU support

Training/Test split: 80% training, 20% testing

A two-layered baseline ANN is trained and CatBoost.

Metrics of Performance: Metrics of Performance used metric synopsis precision predictions that are accurate out of all predictions.

Accuracy: 95%

Precision: 94.7%

Top-3 accuracy: 98.6%

Confusion Matrix

	Predicted: Correct Class	**Predicted: Incorrect Class**
Actual: A-Z	1235(True Positives)	64 (False Negatives)
Other Classes	47 (False Positives)	1195 (True Negatives)

True Positives (TP): 1235

True Negatives (TN): 1195

False Positives (FP): 47

False Negatives (FN): 64

Discussions

The created sign language recognition system performed well on static gesture categorization, with a top-3 accuracy of 98.6% and a validation accuracy of 95%. By successfully extracting hand landmarks, MediaPipe produced clear skeleton images that increased the model's resilience to varying lighting conditions and backgrounds. MobileNetv2 demonstrated accuracy, efficiency, and low latency, making it appropriate for real-time applications.

By letting users remove inaccurate suggestions, particularly for similar gestures like "N" vs "M" and "U" vs "V" vs "W", the option-based prediction interface helped lower errors.

V. CONCLUSION

In this research, we used the MobileNetV2 architecture and a proprietary skeleton-based image dataset to construct a deep learning-based system for static sign language identification. The system was able to achieve excellent classification accuracy while preserving efficiency appropriate for real-time deployment by using MediaPipe for real time manual land mark extraction and training a lightweight CNN model. Usability was further improved by the addition of an option-based prediction interface, which allowed users to reduce categorization errors in visually comparable motions and remove inaccurate predictions. With limited resources, MobileNetV2 offered a dependable and computationally inexpensive solution for real-time sign identification. The model scored a top-3 accuracy of 98.6% and a validation accuracy of 95%, demonstrating high generalization across a range of situations. Despite being restricted to a static motions, the system established a solid framework for further improvements. Future research can concentrate on adding more gesture variations to the dataset, integrating dynamic gesture detection, and investigating sequential models for translation at the sentence level. Furthermore, incorporating this approach with assistive software or mobile platforms may improve accessibility and significantly influence communication assistance for the community of people with hearing impairments.

VI. CHALLENGES

Data Collection Consistency : Maintaining consistency while using a camera to collect data was one of the main problems in this project. The consistency of skeletal images was impacted by changes in hand placement, camera angle, and distance, which may have an effect on model performance.

Gesture Similarity and Misclassification: The skeleton architecture of certain letter signs, such as "M" and "N", "U", "V" and "W" are remarkably similar. Even with a high model accuracy, these commonalities led to sporadic misclassifications. The inability to accurately recognize such motions was further hampered by the absence of contextual information.

Input Features : The method avoided complicated preprocessing procedures like background removal or color-based segmentation by using skeleton-based hand landmark images as input. In addition to ensure consistent input formatting and streamlining the data pipeline, this direct usage of skeletal presentations preserved gesture-specific structural information and enhanced model performance.

VII. FUTURE WORK

Dataset Expansion and Diversity: The model's generalizability can be enhanced by gathering data from several users with various hand shapes and skin tones, as well as by increasing the number of samples per class. Robustness would be further improved by incorporating changes in background, lighting, and camera angles.

Dynamic Gesture Recognition: By adding support for dynamic gestures, future iterations of the system will be able to recognize entire words and phrases. To do this, gesture sequences would need to be captured across time using temporal models like RNNs or LSTMs.

Mobile and Edge Deployment: By tailoring the trained model for embedded devices or mobile platforms, the system can become more useful and accessible for daily usage, particularly in assistive communication aids.

Multilingual Sign Support: By adding support for sign languages from other cultures and geographical areas (such as BSL and ISL), the system's usefulness and influence can be increased for a wider range of international populations.

Gesture-to-Text Translation: By combining a natural language processing module with the recognition system, it is possible to produce grammatically sound phrases from continuous gestures, facilitating more seamless human-computer interaction.

User Personalization: By including lightweight fine-tuning mechanisms, the system can gradually adjust to the unique hand shapes and styles of each user, increasing accuracy.

VIII. ACKNOWLEDGMENT

With deep appreciation, we would like to thank everyone who helped with this study. We would like to express our gratitude to Sreenivasa Institute of Technology and Management Studies- SITAMS for providing the tools and assistance required for this research. We would especially like to thank the professors guided us to complete this project Dr. R. Karunia Krishnapriya, Ms. R. Sathiya Priya for their significant advice and knowledge in the areas of Deep Learning. Their observations greatly improved the Caliber of our work.

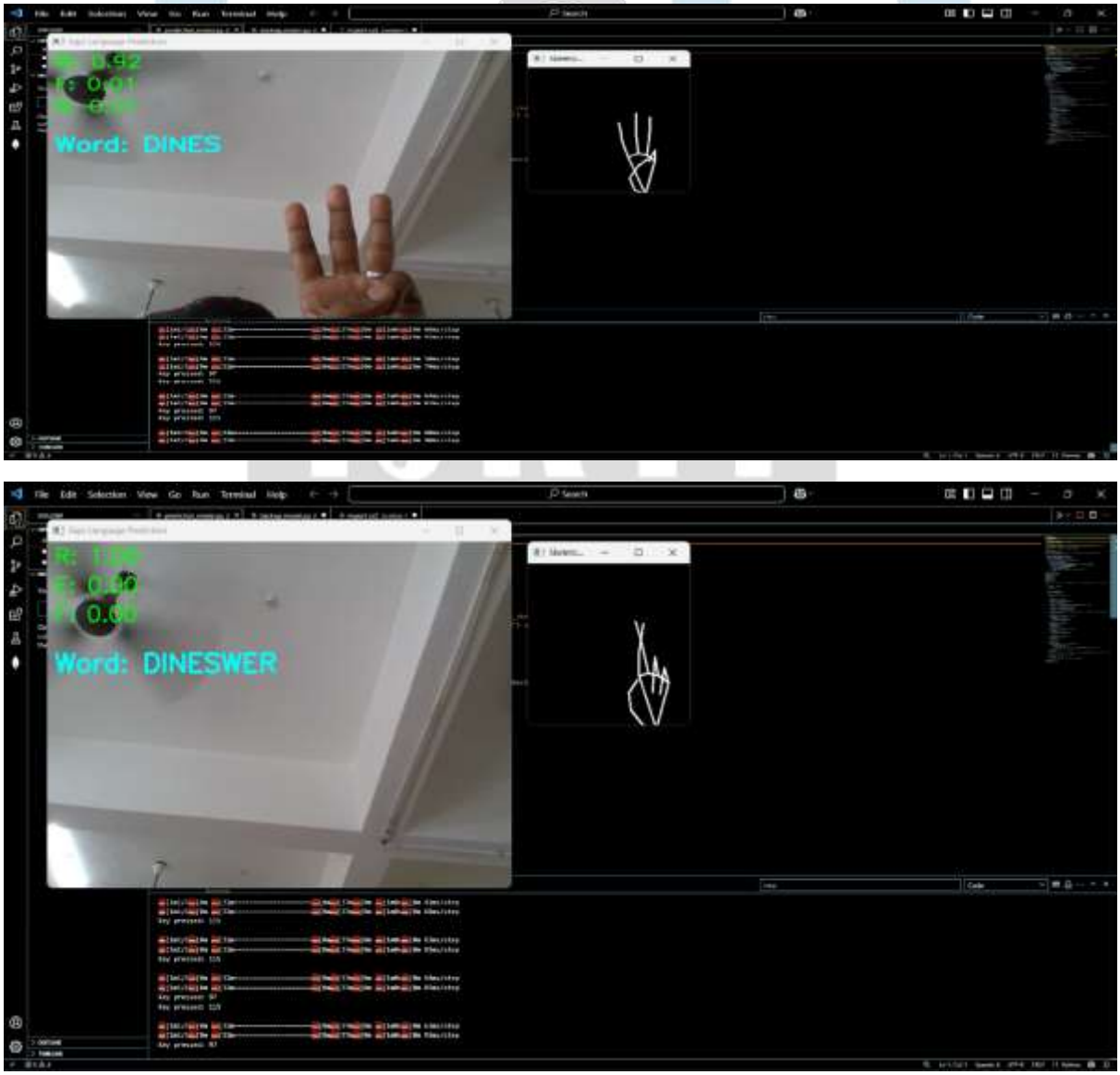

Fig 2. Output

REFERENCES

[1] Dua, D., & Graff, C. (2017). *UCI Machine Learning Repository: Pima Indians Diabetes Dataset*. University of California, Irvine. https://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

[2] Choubey, D., Paul, S., & Pani, S. K. (2020). Early detection of diabetes using artificial neural network. *Materials Today: Proceedings*, 33, 4356–4360.

[3] Tiwari, A. K., & Jha, A. K. (2020). Machine learning based models for early diagnosis of diabetes: A systematic review. *Journal of King Saud University – Computer and Information Sciences*, 34(5), 1708–1721.

[4] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems*, 31.

[5] ivanandam, S. N., & Deepa, S. N. (2007). *Principles of Soft Computing*. Wiley India.

[6] American Diabetes Association. (2022). *Diagnosis and classification of diabetes mellitus*. *Diabetes Care*, 45(Supplement_1), S17–S38.

[7] Krishnan, R., Bhattacharya, S., & Amutha, R. (2018). A novel hybrid feature selection via ANN for diabetes diagnosis. *International Journal of Biomedical Engineering and Technology*, 28(3-4), 252–264.

[8] Misra, R., & Manjunatha, R. (2020). Comparative analysis of classifiers for prediction of diabetes. *Materials Today: Proceedings*, 37, 2966–2971.

[9] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[10] Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/

[11] Nguyen, B. P., Pham, H. V., & Le, N. Q. K. (2022). Machine learning-based approaches for prediction of diabetes mellitus: A review. *Artificial Intelligence in Medicine*, 122, 102204.

[12] Srivastava, S., & Kumar, V. (2021). Comparative study of machine learning algorithms for early detection of diabetes. *Procedia Computer Science*, 185, 43–52.

[13] Rashid, S. M., & Amran, A. (2021). Using CatBoost for interpretable machine learning in medical diagnosis. *International Journal of Advanced Computer Science and Applications*, 12(6), 393–399.

[14] Abiyev, R. H., & Ma'aitah, M. K. S. (2018). Deep convolutional neural networks for chest diseases detection. *Journal of Healthcare Engineering*, 2018.

[15] Zhang, Z., & Zhao, Y. (2021). Application of CatBoost algorithm in diabetes prediction. *Journal of Physics: Conference Series*, 1992(2), 022004.

[16] Chaurasia, V., & Pal, S. (2014). Early prediction of heart diseases using data mining techniques. *Caribbean Journal of Science and Technology*, 2, 208–217.

[17] Kaur, H., & Kumari, V. (2020). Predictive modelling and analytics for diabetes using a machine learning approach. *Applied Computing and Informatics*, 18(1), 90–100.

[18] Zhang, Y., & Li, S. (2020). Real-time diabetes prediction using machine learning models. *IEEE Access*, 8, 207162–207171.

[19] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

[20] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.