International Research Journal of Multidisciplinary Scope (IRJMS), 2024; 5(4):679-688 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2024.v05i04.01241

Enhancing Sign Language Recognition: Leveraging EfficientNet-B0 with Transformer-based Decoding

Rajesh Kumar Singh[1]*, Abhishek Kumar Mishra[1], Ramapati Mishra[2]

[1]Department of Computer Science and Engineering, IFTM University, Moradabad, India, [2]IET, Dr. Rammanohar Lohia Awadh University, Ayodhya (UP), India. *Corresponding Author's Email: rajesh_mtechbu@yahoo.co.in

Abstract

Sign language recognition (SLR) plays a crucial role in facilitating communication for the hearing-impaired community. Conventional methods for SLR have encountered difficulties in attaining both high precision and efficiency because of the intricate characteristics of sign language motions and the variability in articulation. We propose a novel framework for enhancing SLR by leveraging the efficiency of EfficientNet-B0 as a feature extractor and incorporating a transformer-based decoding mechanism for classification. The objective of our method is to enhance the precision and computational effectiveness of SLR systems, thereby making them more viable for real-world applications. Experimental results on two standard, commonly used sign language datasets: American Sign Language (ASL) and ASL with Digits. The proposed model achieves accuracies of 99.59% on the ASL dataset and an outstanding accuracy of 99.97% on the ASL with Digits dataset, outperforming all other state-of-the-art methods. These results highlight the effectiveness of our framework in accurately recognizing sign language gestures, making it highly suitable for real-world applications. Our study contributes to the advancement of SLR research by introducing a novel methodology that combines the efficiency of EfficientNet-B0 with the expressive capabilities of transformer-based decoding, ultimately improving communication accessibility for individuals who rely on sign language.

Keywords: EfficientNet-B0, Multi-Head Self Attention, Sign Language Recognition (SLR), Transformer.

Introduction

Sign language, as a visual-gestural language, plays a crucial role in facilitating communication for those who are deaf or hard-of-hearing. It allows them to effectively convey their thoughts and engage with others. Nevertheless, despite its significance, proficient communication via sign language might prove to be difficult owing to the limited comprehension among the general populace. Sign language recognition (SLR) systems strive to overcome this communication barrier by automatically understanding and converting sign language motions into written or spoken language, therefore increasing communication between those who are deaf and the wider population(1, 2).Traditionally, SLR has often used manual feature extraction techniques, which are then used in conjunction with traditional machine learning algorithms focusing on techniques like Support Vector Machines (SVM) and Hidden Markov Models (HMM). These approaches often rely on handcrafted features and may struggle with capturing intricate details in sign language gestures (3, 4). Although these systems have shown potential, they encounter constraints in dealing with the unpredictability and intricacy of sign language motions, as well as the need for immediate processing in real-world applications. In addition, individuals may have difficulties in applying concepts to various sign languages and adapting to varied styles of articulation. Recent advancements in deep learning, specifically in convolutional neural networks (CNNs) and Transformer architectures, has created new opportunities for speech recognition. Convolutional Neural Networks (CNNs) are very effective in acquiring hierarchical representations from visual data, while Transformers (5) have shown exceptional achievements in problems involving sequence modeling, such as natural language processing and computer vision (2, 6). Using these improvements, we suggest a new framework for improving SLR by merging the efficiency of EfficientNet-B0 with the expressive capabilities of Transformer-based decoding.

(Received 8th June 2024; Accepted 17th October 2024; Published 30th October 2024)

Tan *et al.* proposed the SDViT model, which uses techniques such as transfer learning, fine-tuning of pertained ViTs, early stopping, and knowledge distillation to achieve excellent results in hand gesture recognition. In ensemble learning, stacking distilled student models improves model stability, accuracy, and generalization capabilities. Simplified student models may not fully capture the intricate details found in the original ViT, which could potentially hinder the model's performance (7). Miah *et al.* introduced a neural network that focuses on multistage spatial attention for hand gestures. They also developed a deep learning model that combines feature fusion to improve hand gesture recognition. Nevertheless, the discussion on challenges related to real-time applications is rather concise (8). Kumari and Anand propose a novel hybrid CNN-LSTM framework that focuses on accurately recognizing isolated sign language gestures. They chose the MobileNetV2 backbone model due to its lightweight structure and its capability to extract significant features. Additionally, they optimize the LSTM component with an attention mechanism to selectively focus on important gesture cues. Their method demonstrates an average accuracy of 84.65% on the widely recognized WLASL dataset (9). The study presents a framework consisting of a six-layer Convolutional Neural Network (ConvNet) for feature extraction. The research seeks to address limitations found in existing sign language recognition systems, specifically dealing with low accuracy for certain words caused by similar postures. Additionally, create datasets named BdSL_OPSA22_STATIC1 and BdSL_OPSA22_STATIC2 (10). For sign language recognition, the authors present a specialized multi-headed CNN model. They utilize two input channels, incorporating both images and hand landmarks to ensure reliable data processing. Through the integration of these inputs, the model attains its peak accuracy of 98.98%. Nevertheless, the accuracy experiences a slight decrease to 96.29% when exclusively relying on images, the accuracy drops to 96.29%. The study highlights the impact of real-life application scenarios on model accuracy, emphasizing the presence of noise (11). This study presents the SLRNet-8 architecture, which utilizes Convolutional Neural Networks (CNNs) to recognize American Sign Language (ASL). Combining digits and alphabets results in a slight decrease in the recognition rate. We utilize a range of datasets that encompass ASL gestures, digits, finger spelling, and alphabets for both training and evaluating our models (12). The authors present SASLRM, a system specifically developed to identify Indian Sign Language (ISL) words during emergency situations. An advanced module for selecting key frames improves accuracy by eliminating unnecessary frames. In addition, combining convolution and SA modules improves the performance of SLRS. The study addresses the difficulties associated with self-occlusion in sign language recognition. The model consistently achieves an average accuracy of 95.627 repeated cross-validations (13). EfficientNet-B0 (14) is a pretrained CNN architecture that is extremely lightweight and produces top-notch performance on image classification tasks, all while keeping computational complexity at a minimum. We selected EfficientNet-B0 over alternative architectures like ResNet and Inception due to its superior balance between accuracy and computational efficiency. Through utilizing the pre-trained weights of EfficientNet-B0, researchers can effectively extract distinctive characteristics from photos of sign language. EfficientNet-B0's compound scaling method allows for balanced scaling of network depth, width, and resolution, enhancing its ability to capture intricate features in sign language gestures without incurring excessive computational costs. The extracted characteristics are then inputted into a Transformer-based decoder for the purpose of sequence modeling and classification. This allows the model to effectively represent the intrinsic temporal dependencies and spatial correlations seen in sign language motions. This paper introduces our methods for improving SLR utilizing EfficientNet-B0 with Transformer-based decoding. We assess the effectiveness of our method on commonly used sign language datasets, such as American Sign Language (ASL) (15). The results of our experiment show that our suggested framework performs very well in terms of both accuracy and computing economy, making it highly suitable for real-world applications. In addition, we do ablation experiments to examine the individual contributions of each component in our framework and get a deeper understanding of its efficacy. Our Contribution to the research paper, we proposed EfficientNet-B0 a pretrained CNN module to extract 2D characteristics from sign language images. The Global Average Pooling (GAP) layer is utilized to compress the features from EfficientNet-B0. We have developed a decoding module that utilizes the transformer architecture. The system comprises three components: batch normalization, FFN, and the MHSA mechanism. The paper is structured as follows: Section 2 suggests using EfficientNet-B0 and a multi-head self-attention network to implement sign language recognition. Section 3 presents an in-depth description of the experimental configurations, technical specifications, and metrics used to evaluate the model. Section 4 shows the outcomes of the suggested approaches. Section 5 presents the study's conclusion.

Methodology

Our proposed framework for Sign Language Recognition (SLR) integrates two primary components: feature extraction utilizing EfficientNet-B0 (14) and classification utilizing a Transformer-based decoder (5). The framework is illustrated in Figure 1. In selecting the architecture for our SLR framework, we opted for a combination of EfficientNet-B0 and a Transformer-based decoder due to several key reasons: EfficientNet-B0 offers a highly efficient CNN architecture with significantly fewer parameters and lower computational cost compared to alternatives like ResNet and Inception. Its compound scaling method and inclusion of Squeeze-and-Excitation (SE) blocks enhance its ability to capture intricate features in sign language images, making it ideal for applications requiring real-time processing and deployment on devices with limited resources. The Transformer architecture excels at modeling long-range dependencies and capturing temporal dynamics through its self-attention mechanism. This is crucial for accurately recognizing sign language gestures, which often involve complex spatial-temporal patterns. By integrating EfficientNet-B0's powerful feature extraction with the Transformer's advanced sequence modeling, the model effectively captures both spatial and temporal aspects of sign language gestures. The use of a Global Average Pooling (GAP) layer further optimizes the feature representation for the Transformer decoder. In our approach, we first preprocess the sign language images, which involve resizing them to a fixed size suitable for EfficientNet-B0 input and normalizing pixel values to the range [0, 1]. Additionally, data augmentation techniques such as random cropping and rotation may be applied to augment the dataset. After preprocessing, we leverage the pre-trained weights of EfficientNet-B0 to extract discriminative features from the sign language images. Notably, we incorporate a Global Average Pooling (GAP) (16) layer after EfficientNet-B0 to condense the spatial information across the feature maps, yielding a compact feature representation. These features are subsequently passed to the Transformer-based decoder for classification. We construct a simplified transformer decoding component with N = 3 blocks. Each block has a multi-head self-attention mechanism, an FFN, and batch normalization.

EfficientNet-B0

The Efficient Net architecture represents a notable advancement in pretrained convolutional neural networks (CNNs), particularly distinguished by its adeptness in parameter efficiency and computational speed. Its development incorporates a systematic scaling approach, both simple and compound, to incrementally augment the dimensions of the CNN models, encompassing depth, width, and resolution uniformly. The Efficient Net family consists of seven models, namely EfficientNet-B0 to EfficientNet-B7. This study utilized EfficientNet-B0 to showcase its superiority over ResNet-50 in parameter count and Floating-Point Operations per Second (FLOPs), underscoring its efficacy in feature extraction. In selecting EfficientNet-B0 for our SLR framework, we considered both its computational efficiency and its performance benefits for visual gesture recognition. EfficientNet-B0's architecture, featuring MBConv blocks and SE mechanisms, is adept at capturing fine-grained spatial features essential for distinguishing between similar sign language gestures. Its compound scaling method allows for balanced model scaling, improving the network's capacity to learn intricate patterns without over fitting. With significantly fewer parameters and lower computational demands, EfficientNet-B0 enables real-time processing and deployment on resource-constrained devices,

which is critical for practical SLR applications. Figure 2 presents the schematic representation of the EfficientNet-B0 architecture, arranged into seven blocks according to channel count, striding configurations, and convolutional filter dimensions.

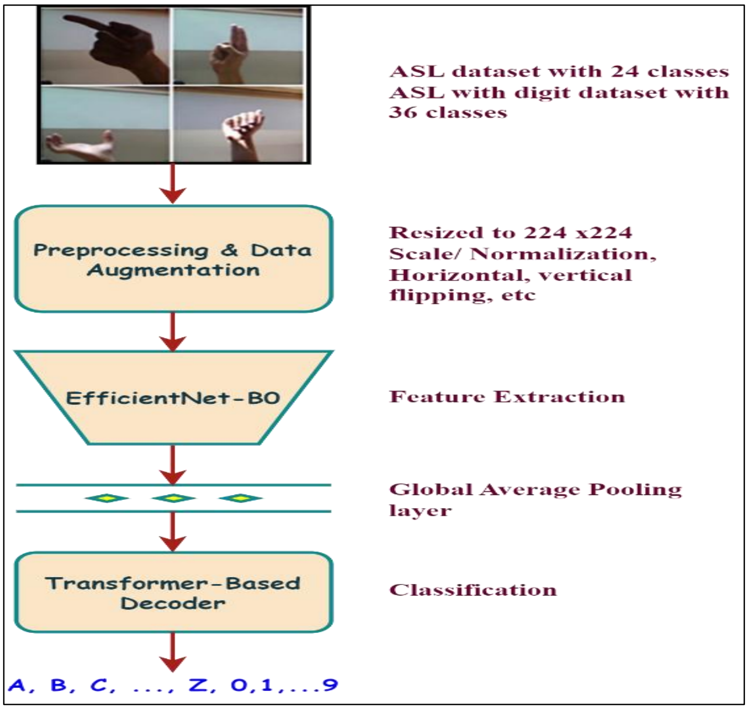
ASL dataset with 24 classes
ASL with digit dataset with 36 classes
Preprocessing & Data Augmentation
Resized to 224 x224
Scale/ Normalization,
Horizontal, vertical flipping, etc
EfficientNet-B0
Feature Extraction
Global Average Pooling layer
Transformer-Based Decoder
Classification
A, B, C, ..., Z, 0,1,...9

Figure 1: Proposed Framework

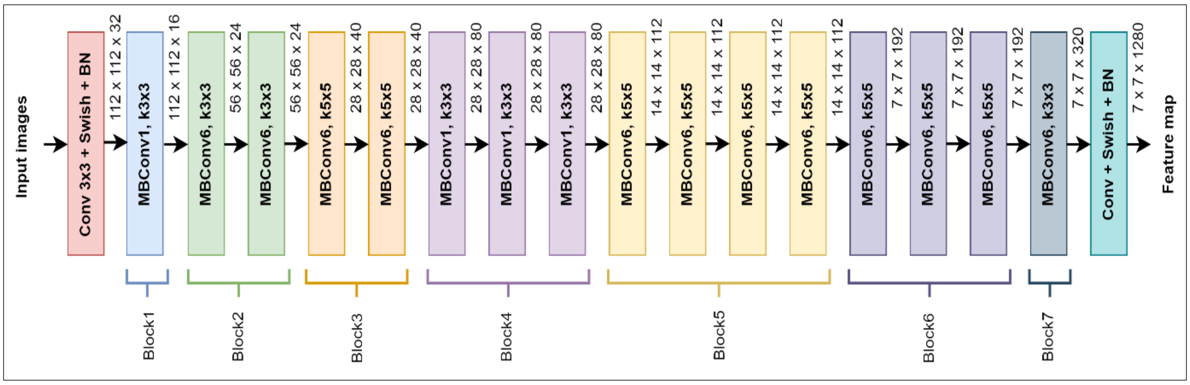

Figure 2: EfficinetNet-B0 Architecture (14)

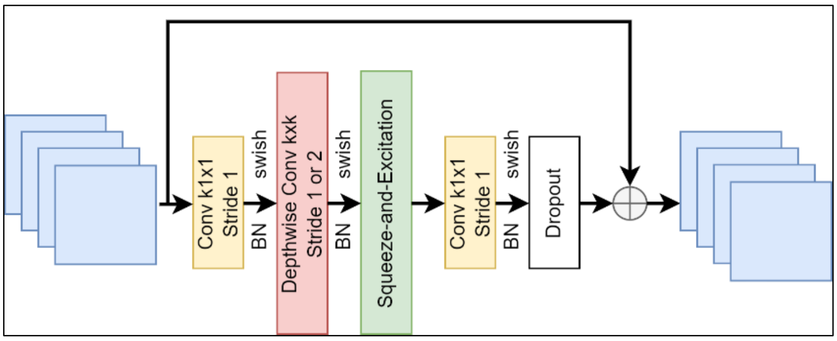

Figure 3: MBConv Model

The core component of EfficientNet-B0 is the mobile inverted bottleneck (MBConv), which is inspired by the MobileNet paradigm. The MBConv architecture is shown in Figure 3. It has a dropout layer, two convolutional layers (k1×1), a depth-wise convolutional layer, and a Squeeze and

Excitation (SE) block. The initial convolutional layer primarily expands the channel, while depthwise convolution minimizes the parameter count. The incorporation of the SE block accentuates the interplay among channels by assigning varying weights to each channel, deviating from uniform allocation. Finally, the subsequent convolutional layer facilitates channel compression. EfficientNet-B0 used Swish activation function (17).

$$f_{swish} = \frac{1}{1 + e^{-\beta x}} \# \qquad [1]$$

Where β is a parameter that can be learned during the training of the CNN. Batch normalization (18), normalizes the output of the convolutional layer to stabilize and speed up training. It applies the following transformation to each feature map:

$$Y = \frac{X - \mu}{\sqrt{\sigma^2 + \epsilon}} * \gamma + \beta \# \qquad [2]$$

Where X is the input feature map, μ and σ are the mean and standard deviation computed over the mini-batch, γ and β are learnable scale and shift parameters, and ϵ is a small constant to avoid division by zero.

Global Average Pooling Layer

The Global Average Pooling (GAP) layer is inserted after the last convolutional layer to reduce spatial dimensions and summarize feature maps.

$$X_{GAP} = GAP\big(EfficientNet - B0(images\ )\big)\#[3]$$

Transformer Based Decoder

The global average pooled feature vector $X_{GAP}$is undergoing a transformer-based decoder for classification. Figure 4, illustrates the transformer-based encoder architecture. The Transformer's self-attention mechanism effectively captures spatial and contextual relationships within the images, allowing the model to weigh the relevance of different parts of the image, enhancing its ability to recognize complex gestures. While our study focuses on static sign language recognition, the Transformer's self-attention mechanism allows the model to capture dependencies within the input data. Even in static images, the self-attention mechanism can model relationships between different parts of the image, effectively capturing spatial dependencies that are crucial for recognizing complex gestures.

Matrices W^q, W^k and W^v with $X_{GAP}{}^i$, to generate three vectors K^i, Q^i and V^i are key, query and value, respectively.

$$Q^i = W^q \cdot X_{GAP}{}^i;\ K^i = W^k \cdot X_{GAP}{}^i;\ V^i = W^v \cdot X_{GAP}{}^i \# \qquad [4]$$

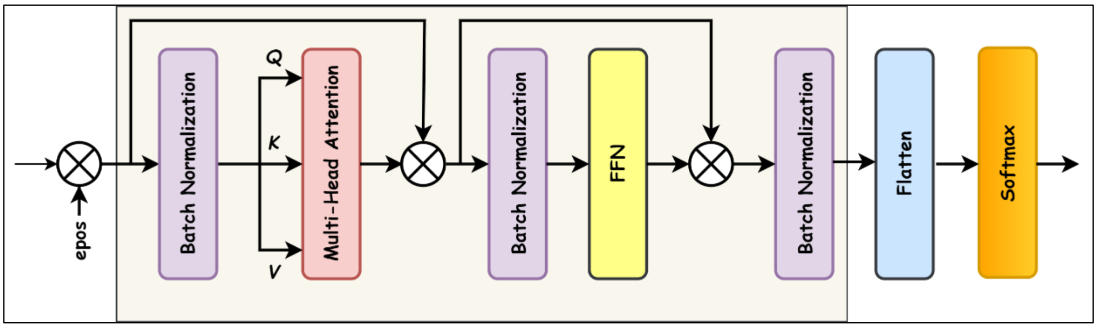

Figure 4: Transformer Based Decoder

The attention mechanism computes the output for each token by first applying the SoftMax function to the scaled dot product of the query and key vectors,Q^i and K^i. This result is then multiplied by the value vectorV^i, producing a refined representation that effectively captures the importance of each token within its context in the input sequence.

$$Attention(Q^i, K^i, V^i) = SoftMax\left(\frac{Q^i \times (K^i)^T}{\sqrt{d_k}}\right) \cdot V^i \#[5]$$

After generating the attention outputs, multiple attention heads' outputs are concatenated through the Multi-Head Self-Attention (MHSA) layer, combining information from different representation subspaces.

$$MHSA = Concat(h_1, h_2, \dots, h_i, \dots, h_H)W^o \# \qquad [6]$$

where, $h_i = Attention\big(Q^i \cdot W_i^Q, K^i \cdot W_i^K, V^i \cdot W_i^V\big)$; Our model incorporates three such attention heads, and the resulting output from the attention layer has a dimensionality of 512.The concatenated output from the MHSA layer is then passed into a position-wise feed-forward network

(FFN). This network consists of two sequential linear transformations, with a non-linear activation function (GeLU) applied between them.

$$FFN(x) = \sigma(xW_1 + b_1)W_2 + b_2 \# \quad [7]$$

where, W_1, W_2denote the weights of the feed-forward network,$x$corresponds to the output from the MHSA layer, represented as$Output_{MHSA} = MHSA^1, \dots, MHSA^i, \dots, MHSA^n$. The parameter$b_1 b_2$represent biases, while σrefers to the $GeLU(\cdot)$ transfer function. After processing through the FFN, the output is subjected to a normalization layer, culminating in the final output of the single-layer encoder.

$$y_{transformer} = flatten\big(FFN(x)\big)\# \quad [8]$$

In the final stage, a SoftMax activation function is employed to predict the probability distribution of each token$y_i$in the response, determining the most likely outcomes based on the refined representations created by the preceding layers.

$$p(y_0, \dots, y_{i-1}) = SoftMax\big(W \cdot y_{transformer}\big)\# \quad [9]$$

This architecture leverages the self-attention mechanism's ability to weigh the relevance of different tokens in a sequence, allowing for a more nuanced understanding of the input. The model can capture more prosperous relationships within the data by incorporating multiple attention heads and combining their outputs. The subsequent feed-forward network and normalization steps refine these representations, ensuring a comprehensive understanding of the input sequence informs the final predictions (19). To balance model complexity and performance, we employed three Transformer layers in our architecture. We set the number of attention heads to three, which allows the model to capture information from different representation subspaces effectively. The dimensions of the query, key, and value vectors were configured to be 64. We utilized the Gaussian Error Linear Unit (GeLU) activation function in the Feed-Forward Network (FFN) due to its smooth and non-linear properties. To prevent overfitting, a dropout rate of 0.2 was applied.

Experimental Setups

Datasets

This section provides a comprehensive overview of the datasets employed for performance evaluation. The ASL dataset consists of 24 classes that capture alphabet gestures from A to Y, excluding J and Z. In addition, the ASL dataset comprises 36 classes, consisting of 26 alphabets and 10 digits (15). Table 1 presents a succinct overview of the aforementioned datasets.

Implementation Details

We employ the EfficientNet-B0 pretrained CNN model for feature extraction and a transformer-based decoder for classification. We used categorical cross-entropy loss as the training method, and the Adam optimizer carried out weight updates. This optimizer enhances model performance by fine-tuning weight values. To optimize the learning rate, the training involved 100 epochs and incorporated a learning rate scheduler. We set the dropout ratio at 0.3 to address over fitting issues. Table 2 presents a comprehensive summary of the hyper parameters. We employ data augmentation as a solution for addressing the issue of class imbalance.

Table 1: Summary of the Datasets

Datasets	Number of Classes	Total Samples
ASL	24	65774
ASL with digits	36	2515

Table 2: Training Parameter List of Transformer-Based Encoder

Parameters	Values
Learning rate	0.0001
Batch size	64
Epoch	100
Dropout	0.3
Optimizer	Adam
Scheduler	Learning rate scheduler
Loss function	Categorical cross entropy

We conduct the experiments using the Linux-Mint Cinnamon Operating System. We execute the training process of the proposed model on a high-performance system featuring an i7 processor, 32 GB of RAM, and an 8GB NVIDIA GTX 4060 GPU. We implement the proposed model using Python.

Results and Discussion

Ablation Study

The effectiveness of transformer-based encoder in sign language recognition methods. We evaluate the efficacy of sign language recognition techniques using the transformer-based encoder, both with and without it. The approaches utilize multiple pretrained Convolutional Neural Networks (CNN), such as VGG16 (20), ResNet-50 (21), MobileNet (22) and EfficientNet-B0 (14). Table 3 displays the accuracy results for both the ASL dataset and the ASL dataset with digits. Table 3, shows that, when it comes to the ASL dataset, incorporating transformer layers significantly improved accuracy for various model architectures. Notably, VGG16 demonstrated a remarkable increase of 2.75% in accuracy. Similarly, ResNet-50 and MobileNet showed significant percentage improvements of 1.16% and 1.04%, respectively. The EfficientNet-B0 model, known for its high performance, demonstrated a significant increase of 2.63% in accuracy. This finding confirms the effectiveness of transformer layers in improving the model's ability to interpret ASL gestures. The ASL with Digits dataset highlights the effectiveness of incorporating transformers to enhance accuracy metrics. We found that transformers significantly improve accuracy across various architectures, including VGG16, ResNet-50, MobileNet, and EfficientNet-B0. In terms of accuracy, VGG16 showed a significant increase of 2.68%, while ResNet-50 and MobileNet saw improvements of 1.65% and 1.38%, respectively. There was a noticeable improvement of 1.69% in accuracy, even within the already proficient EfficientNet-B0 framework. These findings highlight the importance of transformer architectures in effectively handling the challenges posed by simultaneous digit recognition and ASL gestures. This enables a more accurate understanding of combined sign language and numerical expressions in real-world situations.

The comprehensive evaluation of our model's performance, as depicted in Table 4, demonstrates that integrating the Transformer-based decoder with EfficientNet-B0 leads to significant improvements across all performance metrics on both the ASL and ASL with Digits datasets. Specifically, the accuracy on the ASL dataset increased from 97.23% to 99.59% and on the ASL with Digits dataset from 97.21% to an impressive 99.97%, indicating a substantial enhancement in the model's overall ability to correctly classify sign language gestures. Precision improved from 97.84% to 99.24% on the ASL dataset and from 97.91% to 99.36% on the ASL with Digits dataset, reflecting a notable reduction in false positives and demonstrating the model's enhanced capability to correctly identify relevant gestures without misclassification. Similarly, recall increased from 97.31% to 99.11% and from 97.18% to 99.07% on the respective datasets, indicating the model's improved proficiency in capturing all pertinent instances of sign language gestures, thus reducing missed detections.

Table 3: Baseline Model with and Without Transformer-Based Decoder

Models	Accuracy	
	ASL	ASL with Digits
VGG16	82.67	83.01
VGG16 + Transformer	86.21	88.50
ResNet-50	81.42	82.53
ResNet-50 + Transformer	85.52	87.11
MobileNet	88.69	88.21
MobileNet + Transformer	92.68	93.54
EfficientNet-B0	97.23	97.21
EfficientNet-B0 + Transformer	99.59	99.97

Table 4: Performance Metrics on ASL and ASL with Digits Dataset

Model	ASL			ASL with Digits		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
EfficientNet-B0	97.84	97.31	97.57	97.91	97.18	97.54
EfficientNet-B0 + Transformer	99.24	99.11	99.17	99.36	99.07	99.21

The F1-Score, which harmonizes precision and recall, rose from 97.57% to 99.17% on the ASL dataset and from 97.54% to 99.21% on the ASL with Digits dataset, underscoring a balanced and significant improvement in overall classification performance. These consistent enhancements across all metrics underscore the efficacy of the Transformer-based decoder in capturing complex spatial dependencies and contextual nuances inherent in sign language images. The self-attention mechanism within the Transformer enables the model to focus on the most salient features, enhancing its discriminative capabilities and robustness against variations in signing styles and environmental conditions. Consequently, the integration of EfficientNet-B0 with the Transformer-based decoder not only achieves higher accuracy but also demonstrates enhanced reliability and efficiency, highlighting its potential for developing practical and robust sign language recognition systems suitable for real-world applications that demand high precision and recall.

Comparison with Other State-of-Art Methods

Our proposed methodology is highly accurate on both datasets, setting it apart from other approaches. Although traditional approaches like LBP + PNN achieve decent accuracy, utilizing deep learning models like CNN and ADCNN produces remarkable results. For instance, on the ASL dataset, CNN achieves an accuracy of 99.78% and ADCNN achieves 98.50%. Similarly, on the ASL with Digits dataset, CNN achieves an accuracy of 98.65% and ADCNN achieves 98.49%. In addition, when transformer architectures are integrated into CNN models, such as in ViT with lightweight CNN and multi-head CNN, they are able to achieve impressive accuracies of 98.17% and 98.98%, respectively. Nevertheless, our model, which combines EfficientNet-B0 with Transformer, outperforms all of them. It achieves impressive accuracies of 99.59% on the ASL dataset and an exceptional 99.97% on the ASL with Digits dataset, showcasing its superiority over other methods in the comparison, show in Table 5.

Table 5: Comparison with Other State-of-Arts Method on ASL and ASL with Digit Dataset

Models	Accuracy (%)	
	ASL	ASL with Digits
LBP + PNN (4)	93.33	-
CNN (23)	99.78	98.65
ADCNN (24)	98.50	98.49
ViT + Lightweight CNN (2)	98.17	
Multi-head CNN (11)	98.98	
EfficientNet-B0 + Transformer (Ours)	**99.59**	**99.97**

Practical Implementation Challenges

Environmental Unpredictability: Variations in lighting conditions, backgrounds, and camera angles can affect model performance. To mitigate this, we employed data augmentation techniques during training, such as random brightness adjustments, rotations, and translations, to enhance the model's robustness. Future work includes exploring domain adaptation methods to further improve performance in diverse environments.

Signer Diversity: Differences in hand shapes, sizes, skin tones, and individual signing styles pose challenges. Expanding the training dataset to include a diverse set of signers and utilizing transfer learning can help the model generalize better. We also suggest exploring personalized models or adaptive algorithms that can adjust to individual users over time.

Real-Time Processing: Our model is designed to be computationally efficient, leveraging EfficientNet-B0 and an optimized Transformer decoder. This efficiency allows for real-time processing, which is critical for practical applications. Potential optimizations include model quantization and pruning, and the use of hardware accelerators like GPUs and TPUs. Future work will focus on deploying the model on edge devices and validating real-time performance.

Conclusion

This study presents a robust framework for Sign Language Recognition (SLR) that integrates EfficientNet-B0 with a Transformer-based decoder. The method outperforms existing methods on both the ASL and ASL with Digits datasets, thanks to deep learning advancements. The inclusion of transformer layers enhances accuracy across different model architectures, demonstrating the effectiveness of this approach in improving SLR models' interpretive capabilities. However, the reliance on pre-trained models may limit the framework's adaptability to diverse sign languages and gestures. Despite these limitations, the model achieved high accuracy rates of 99.59% and 99.97% on the ASL and ASL with Digits datasets; we acknowledge challenges in practical implementation, including environmental unpredictability, signer diversity, and real-time processing requirements. Addressing these challenges is crucial for deploying SLR systems effectively. Future research will focus on optimizing and expanding the framework, including developing techniques for adapting to new sign languages, improving efficiency for real-time deployment on edge devices, and enhancing the model's robustness to environmental variability and signer diversity.

Abbreviations

FFN: Feed Forward Networks, MHSA: Multi-head Self Attention, GAP: Global Average Pooling, ASL: American Sign Language, BN: Batch Normalization, Q^i, K^i, V^i: Query, Key, Values.

Acknowledgement

Nil.

Author Contributions

Rajesh Kumar Singh: Conceptualization, methodology, data collection, data analysis, original draft preparation. Abhishek Kumar Mishra: Supervision, formal analysis, validation. Ramapati Mishra: Supervision, formal analysis, method validation.

Conflict of Interest

The authors declare no conflicts of interest.

Ethics Approval

Not applicable.

Funding

Nil.

References

1. Deafness and hearing loss, World Health Organization. 2024.. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.
2. Zhang Y, Wang J, Wang X, Jing H, Sun Z, Cai Y. Static hand gesture recognition method based on the Vision Transformer. Multimedia Tools and Applications. 2023 Aug;82(20):31309-28.
3. Gajalakshmi P, Sharmila TS. Hand gesture recognition by histogram based kernel using density measure. In: 2019 2nd International Conference on Power and Embedded Drive Control (ICPEDC). 2019; 294–8.
4. Sadeddine K, Djeradi R, Chelali FZ, Djeradi A. Recognition of Static Hand Gesture. In: 2018 6th International Conference on Multimedia Computing and Systems (ICMCS). 2018; 1–6.
5. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is All You Need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc. 2017; 6000–10.
6. Adithya V, Rajesh R. A deep convolutional neural network approach for static hand gesture recognition. Procedia computer science. 2020 Jan 1;171:2353-61.
7. Tan CK, Lim KM, Lee CP, Chang RKY, Alqahtani A. SDViT: Stacking of Distilled Vision Transformers for Hand Gesture Recognition. Appl Sci. 2023 Nov 10;13(22):12204.
8. Miah AS, Hasan MA, Shin J, Okuyama Y, Tomioka Y. Multistage spatial attention-based neural network for hand gesture recognition. Computers. 2023 Jan 5;12(1):13.
9. Kumari D, Anand RS. Isolated Video-Based Sign Language Recognition Using a Hybrid CNN-LSTM Framework Based on Attention Mechanism. Electronics. 2024 Apr 1;13(7):1574.
10. Rahaman MA, Oyshe KU, Chowdhury PK, Debnath T, Rahman A, Khan MS. Computer vision-based six layered convneural network to recognize sign language for both numeral and alphabet signs. Biomimetic Intelligence and Robotics. 2024 Mar 1;4(1):100141.
11. Pathan RK, Biswas M, Yasmin S, Khandaker MU, Salman M, Youssef AAF. Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network. Scientific Reports. 2023 Dec 1;13(1):14381.
12. Rahman MM, Islam MS, Rahman MH, Sassi R, Rivolta MW, Aktaruzzaman M. A new benchmark on American Sign Language recognition using convolutional neural network. In: 2019 International Conference on Sustainable Technologies for Industry 40, STI 2019. Institute of Electrical and Electronics Engineers Inc;

2019.
13. Das S, Biswas SKR, Purkayastha B. An Expert System for Indian Sign Language Recognition Using Spatial Attention–based Feature and Temporal Feature. ACM Transactions on Asian and Low-Resource Language Information Processing. 2024 Mar 9; 23(3):1-23.
14. Tan M, Le Q V. EfficientNet: Rethinking model scaling for convolutional neural networks. 36th Int Conf Mach Learn ICML. 2019 June:6105-6114.
15. American Sign Language Dataset. https://www.kaggle.com/datasets/kapillondhe/american-sign-language
16. Patel K, Wang G. A discriminative channel diversification network for image classification. Pattern recognition letters. 2022 Jan 1; 153:176-82.
17. Ramachandran P, Zoph B, Le QV. Searching for activation functions. arXiv preprint arXiv:1710.05941. 2017 Oct 16.
18. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning -JMLR.org. 2015; 37: 448–56. (ICML'15).
19. Kumar H, Dwivedi A, Mishra AK, Shukla AK, Sharma BK, Agarwal R, et al. Transformer-based decoder of melanoma classification using hand-crafted texture feature fusion and Gray Wolf Optimization algorithm. MethodsX. 2024; 13:102839.
20. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR 2015). 2015. p. 1–14.https://arxiv.org/abs/1409.1556
21. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; 770-778.
22. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. MobileNets: efficient convolutional neural networks for mobile vision application. arXiv preprint arXiv:1704.04861. 2017 Apr;126.
23. Ahuja R, Jain D, Sachdeva D, Garg A, Rajput C. Convolutional Neural Network Based American Sign Language Static Hand Gesture Recognition. Int J Ambient Comput Intell. 2019 Jul; 10(3):60–73.
24. Alani AA, Cosma G, Taherkhani A, McGinnity TM. Hand gesture recognition using an adapted convolutional neural network with data augmentation. In: 2018 4th International Conference on Information Management (ICIM). 2018:5–12.