IEEE Open Journal of the
Computer Society

Received 5 February 2024; accepted 21 February 2024. Date of publication 28 February 2024;
date of current version 18 April 2024. The review of this article was arranged by Associate Editor Zhi Liu.

Digital Object Identifier 10.1109/OJCS.2024.3370971

Hand Gesture Recognition for Multi-Culture Sign Language Using Graph and General Deep Learning Network

**ABU SALEH MUSA MIAH [1], MD. AL MEHEDI HASAN [2] (Member, IEEE),
YOICHI TOMIOKA [1] (Member, IEEE), AND JUNGPIL SHIN [1] (Senior Member, IEEE)**

[1]School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan
[2]Department of Computer Science & Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh

CORRESPONDING AUTHOR: JUNGPIL SHIN (e-mail: jpshin@u-aizu.ac.jp).

This work was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP23H03477.

ABSTRACT Hand gesture-based Sign Language Recognition (SLR) serves as a crucial communication bridge between hard of hearing and non-deaf individuals. The absence of a universal sign language (SL) leads to diverse nationalities having various cultural SLs, such as Korean, American, and Japanese sign language. Existing SLR systems perform well for their cultural SL but may struggle with other or multi-cultural sign languages (McSL). To address these challenges, this paper introduces a novel end-to-end SLR system called GmTC, designed to translate McSL into equivalent text for enhanced understanding. Here, we employed a Graph and General deep-learning network as two stream modules to extract effective features. In the first stream, produce a graph-based feature by taking advantage of the superpixel values and the graph convolutional network (GCN), aiming to extract distance-based complex relationship features among the superpixel. In the second stream, we extracted long-range and short-range dependency features using attention-based contextual information that passes through multi-stage, multi-head self-attention (MHSA), and CNN modules. Combining these features generates final features that feed into the classification module. Extensive experiments with five culture SL datasets with high-performance accuracy compared to existing state-of-the-art models in individual domains affirming superiority and generalizability.

INDEX TERMS SL, SLR, GCN, MHSA, McSL, Graph meets with attention and CNN (GmTC), Hand gesture recognition (HGR), Bangla sign language (BSL), Korean sign language (KSL), American sign language (ASL), Japanese sign language (JSL).

I. INTRODUCTION

Hand gesture recognition is a crucial aspect of Human-Computer Interaction (HCI) and computer vision, especially in applications like SLR, facilitating nonverbal communication between hard of hearing and non-deaf communities. Hand gestures, integral to daily activities, convey specific information through hand orientation, posture, and distinct movements. They symbolize letters, digits, or objects, often relying on hand orientation for meaning, and some gestures having universal meanings and others varying based on culture or context [1], [2], [3]. SL is the only communication medium among the hard of hearing and non-deaf communities. However, the hard of hearing community faces ongoing challenges to meet their basic needs in the modern digital era.

Globally, the World Health Organization (WHO) reports 466 million people with deafness or disabling hearing loss [4]. In the U.S., the National Institute on Deafness estimates 15% of adults (37.5 million) experience difficulty hearing [5]. South Korea reports around 1.6 million individuals with hearing impairments [6], and Japan notes approximately 370,000 people with hearing impairments [7]. Bangladesh and other countries also host significant hard of hearing communities.

The global diversity of SLs, including American Sign Language (ASL), Korean Sign Language (KSL), Bengali Sign Language (BSL), Japanese Sign Language (JSL), and Arabic Sign Language (ArSL), Large Scale Argentine Sign Language (LSA64) poses challenges for cross-cultural communication [8], [9], [10]. While the hard of hearing community is eager to learn sign language through special language schools, the non-deaf community often lacks interest. This situation necessitates costly human SL translators. To address these challenges, researchers have been working to develop automatic multi-cultural SLR systems using machine learning and deep learning approaches [8], [9], [10], [11]. Vision-based SL datasets have become popular due to their portability and cost-effectiveness over the sensor-based system [8], [10], [12]. Traditional approaches involved hand-crafted features and machine learning algorithms [1], [8], [13]. However, these face challenges with large-scale datasets and diverse backgrounds. To overcome the challenges, researchers have turned to end-to-end deep learning-based Convolutional Neural Networks (CNN) for SLR [9], [10], [14].

Miah et al. achieved 93.00% and 99.00% accuracy for BdSL38 and ASL datasets with the CNN-based BenSignNet model [10]. Some researchers combined self-attention models with CNN to enhance accuracy and generalizability compared to only CNN model [13], [15]. Transformer models like ViT, IPT, and SETR also have been employed [16], [17], [18], [19] to improve the performance accuracy and generalizability. However, fixed patch sizes are the most highlighted challenges in these models that are addressed by the CNN meeting Transformer (CMT) model [20]. Shin et al. enhanced CMT to improve the performance accuracy of the KSL, and they reported 89.00% accuracy for KSL-77 and 98.00% for KSL-20 [21], which is 10% high accuracy compared to the previous model [22]. However, among the mentioned SLR systems, Various technologies were employed for specific culture-based SLR systems, e.g., KSL [21], [22], [23], ASL [1], [13], [21], [24], [25], [26], [27], [28], BdSL [10], [29], [30], and JSL [7].

Existing SLR systems often excel in specific cultural contexts but struggle with others. For instance, BenSignNet [10] may face challenges in recognizing KSL, ASL, or JSL, and modified CMT [21] may encounter difficulties with BdSL, JSL, or ASL datasets, and vice versa. The main drawback of these systems is that they can perform well in specific cultural contexts of SL and struggle with other SLs. For example, BenSignNet [10] may face challenges in recognizing KSL, ASL, or JSL, and modified CMT [21] encounter difficulties in getting satisfactory performance with BdSL, JSL, or ASL datasets, and vice versa. Nurnoby et al. addressed cultural dependencies in SLR with a CNN-based multi-cultural SLR system [11]. However, its limitations include an evaluation with only two datasets (ArSL and ASL) and suboptimal performance due to ineffective features, hindering real-time deployment for McSL recognition.

In light of this situation, there is a crucial need for an effective McSL to assist hard of hearing and speech-impaired communities in various scenarios, such as interacting with individuals whose nationalities are unknown. Our study introduces an advanced McSL system, incorporating insights from superpixel technology [31], a GCN network, and modified CMT [20] module with MHSA. The primary contributions of our study are outlined below:

- *Novelty:* We introduce an innovative graph meeting with the Transformer and CNN (GmTC) Model for Multi-Culture Sign Language (McSL) recognition, fusing graphs, and general deep neural network (DNN) based features. Our novel approach leverages superpixel-based GGCN, Multi-Head Self-Attention (MHSA), and deep learning layers to extract highly effective features, establishing the McSL recognition system. By seamlessly integrating short-range and long-range dependencies from GCN, deep learning layers, and MHSA, the GmTC Model offers a pioneering solution in SLR technology.
- *Adaptive Feature Aggregation with Dual Streams:* GmTC employs an adaptive approach that integrates graph-based and general deep learning feature aggregation through two parallel streams: GCN and general deep learning streams. In the GCN stream, we introduce a pioneering use of Simple Linear Iterative Clustering (SLIC) for superpixel partitioning, transforming them into a fully connected graph. This leverages spatial relationships among superpixels to extract effective features, introducing a groundbreaking distance-based pixel relationship feature using GCN.
  The second stream focuses on attention-based features, undergoing multi-stage processing through MHSA and CNN modules. A dedicated grain module addresses fixed-size patch challenges, facilitating the extraction of multiscale features. The subsequent combination of MHSA and CNN stages captures long-range and short-range pixel dependencies, setting a new standard for feature extraction in the field.
- *Innovative Adaptation and Generalization:* The GmTC model stands out with adaptive learning, surpassing traditional feature expressions for superior generalization. Extensive evaluations across SLR datasets (KSL, JSL, BSL, ASL, LSA64) demonstrate GmTC's high-performance accuracy, surpassing benchmarks set by both high-performance CNNs and canonical transformers. GmTC achieves this by seamlessly integrating GCN features, DNN features, and long-range dependencies from MHSA. This unique combination elevates performance accuracy and generalizability, making GmTC a sophisticated and efficient solution for a McSL. Our explanation of the portion of the code can be found in the following link: https://github.com/musaru/GmTC

We organize the rest of the paper as follows: A literature review is described in Section II, in Section III we included various hand gestures, and SL datasets Section IV describes the architecture of the proposed system. Section V highlight the evaluation performance. In Section VII, draw the conclusion and future work.

II. RELATED WORK

Researchers employed various technologies to develop the hand gesture-based SLR system, specifically hand-crafted feature extraction with machine learning algorithms and deep learning algorithms [19], [32]. Many researchers extracted hand-crafted features and employed machine learning algorithms such as the Hidden Markov model (HMM), [1], Pattern Trees (SP-Tree) [33] and they reported 93.00% accuracy for Greek Sign Language (GSL) and 88.00% accuracy for German Sign Language (GSL) respectively. Sequentially, Linear Discriminant Analysis (LDA), k-nearest Neighbors (KNN), and Random Decision Forest (RDF) also proved their efficiency for various SL datasets [19].

Researchers focus on deep learning models for effective, generalized hand gesture recognition with large-scale datasets that face difficulties in machine learning algorithms. Miah et al. utilized a CNN, the BenSignNet model, achieving 93.00% and 99.00% accuracy for the BdSL38 and ASL datasets, respectively [10]. Similarly, deep learning has proven ability in CSL, ASL [34] and Arabic sign langauge [35]. Most of the mentioned systems can produce good accuracy for the specific cultural SL, but they may face difficulties with other or multi-cultural SLs. To overcome the challenges, researchers focus on transfer learning, including VGGNets [36] and AlexNet [37], InceptionNet and GoogLeNet [38], ResNet [39]. This transfer learning has been acknowledged as a valuable technique, encompassing various methods to leverage pre-trained models for task-specific accuracy improvement.

Recently, some researchers achieved good performance in vision-based hand gesture recognition using the Vision Transformer (ViT) [40], [41] is prominent in hand gesture recognition for SL applications [41]. The ViT transformer, utilizing only a patch of the image, can result in potential information loss. Guo et al. introduced a transformer model, CNN meets Transformer (CMT), by incorporating self-attention with CNN layers to efficiently extract multi-scale features [20]. Shin et al. further optimized CMT and reported 89.00% and accuracy for KSL-77 and for KSL-20 respectively [21], [22].

However, among the mentioned SLR systems, Various technologies were employed for specific culture-based SLR systems, e.g., KSL [21], [22], [23], ASL [1], [13], [21], [24], [25], [26], [27], [28], BdSL [10], [29], [30], and JSL [7].

Existing SLR systems often excel in specific cultural contexts but struggle with others or McSL. To address the challenges, we introduce an advanced McSL system, incorporating insights from superpixel technology [31], Graph Convolutional Neural (GCN) network, and a CMT module with MHSA.

TABLE 1. Dataset Description

Dataset Names	Language	Signs	Total Sample	Sample Sign	Type
KSL-77	Korea	77	112,564	1461	SL
KSL-20	Korea	20	96200	4800	SL
BSL	Bangla	38	12160	320	SL
Lab BSL	Bangla	38	22800	600	SL
ASL-10	America	10	2800	120	SL
ASL-20	America	20	18000	900	SL
JSL	Japense	41	7380	1800	SL
Creative Senz3d	General	11	1320	300	Digit
NTU Dataset	Digits	10	1000	100	Common Gesture

FIGURE 1. Sample images of the KSL-20 dataset.

III. DATASETS

We evaluated the proposed GmTC model using different SL datasets as multi-culture sign language recognition systems (McSL), including JSL, KSL, ASL, BSL and LSA64 datasets. The dataset we utilized here to evaluate the model is demonstrated in Table 1.

A. KSL DATASET

KSL is among the most widely used languages globally, and the KSL-77 and KSL 20 datasets are utilized in the study for evaluation [21], [22]. The KSL-77 dataset, which was collected from 20 individuals and includes 1,229 videos, from which 112,564 frames were extracted at a rate of 30 frames per second [22]. KSL-20 is another famous dataset for the KSL, which consists of 20 videos, and the recordings mainly consist of 4-second videos, with two repetitions for each sign from each signer [21]. Fig. 1 provides an example of a KSL word dataset beside Table 1.

B. ASL DATASET

We evaluated the proposed model with ASL-10 and ASL-20 datasets, and this dataset mainly focuses on fundamental hand gestures commonly used worldwide [12]. ASL-10 comprises ten distinct gestures from 14 individuals, with ten instances of each gesture, producing 1400 unique data samples. Another famous dataset is ASL-20, which consists of 20 ASL words and is composed of 18000 frames in total. Fig. 2 demonstrated the sample images of this dataset.

C. BANGLA SIGN LANGUAGE (BSL) DATASET

Bangla is the 3rd most widely used language globally, and Bangladesh has a population of 3 million individuals

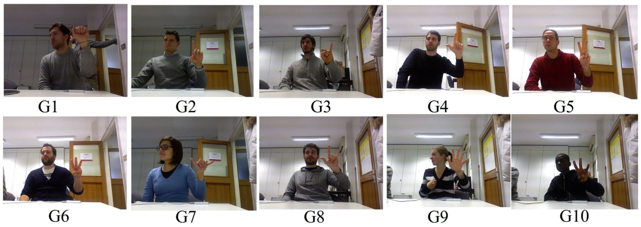

FIGURE 2. Sample image of the ASL sign word dataset.

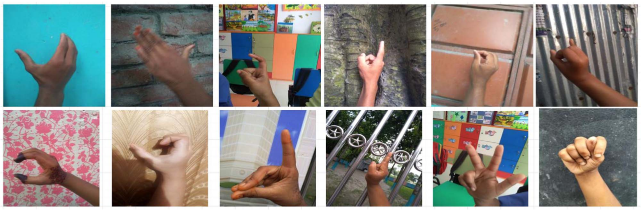

FIGURE 3. Example of BSL image from our lab BSL dataset.

with hearing and speech impairments. Despite its significance, there are limited publicly available datasets for Bangla SLR [42]. Due to the scarcity of datasets for BSL, we also curated a new dataset within our lab. For the Bangla Sign Language (BSL) dataset selection, we focused on commonly used alphabet gestures that convey comprehensive information, referring to 'The Bangla Sign Language Dictionary' by the National Centre for Special Education. Collaborating with Proyash, Rangpur, a special education school for the deaf-mute community, we recorded the dataset with guidance from a SL instructor. The dataset comprises 38 gestures representing the Bangla alphabet from the National Federation, depicted in Fig. 3. It includes contributions from both general and hearing-impaired individuals, with 15 participants—10 students from Bangladesh Army University of Science and Technology (BAUST) and five hard of hearing students from Proyash School, Rangpur. Table 1 provides participant details. With over 600 samples for each of the 38 classes, totalling around 22,800 samples, the size of each sample is 512 × 512 pixels, as illustrated in Fig. 3. In addition, we also tested the model with the existing BSL-38 dataset that consists of 38 classes [29]. Each class encompasses 320 images, resulting in 12,160 images across the 38 classes. The dataset creation involved 42 deaf students and 278 non-deaf students.

D. JSL DATASET

The JSL dataset encompasses the 41 Japanese sign characters, which are composed of the RGB image, and the sample size has been adjusted to 400 × 400 and comprises 7,380 images, encompassing 180 samples per class. These images were captured from 18 individuals, with ten images per person.

E. LSA64 DATASET

We also evaluated the proposed model with a benchmark Large Scale Argentinian Sign Language (LSA), consisting of 3200 videos with the participation of 10 non-expert subjects, each executing 5 repetitions of 64 unique sign types. The chosen signs represent frequently used expressions within the LSA lexicon, covering verbs and nouns [43].

F. HAND GESTURE DATASET

We also assessed the proposed model using digit and hand gesture datasets, specifically the NTU Dataset [32] and Senz3D [44]. NTU dataset consists of ten gestures representing decimal digits 0 to 9, recorded from 10 individuals; the dataset includes 1000 images with a resolution of 640 × 480. We also evaluated our model with the Senz3d dataset comprising 11 unique hand gestures. Every gesture from each person has been replicated 30 times, leading to a total collection of 1320 instances. The collection includes 3200 samples for every gesture, and each RGB image adheres to a 640×480 resolution.

IV. PROPOSED METHODOLOGY

Fig. 4 demonstrated the architecture of the proposed model. The study mainly aims to make a generalized system for a multi-culture sign language recognition system (McSL) using graphs and a general DNN. The RGB image can be written as $Input_{SingleImage} = X_R^i$, where $X_R^i \in R^{(M\times N\times C)}$ M = 90, N = 90 and C = 3 indicate width and height and channel, respectively.

We proposed Graph meets with Attention and CNN (GmTC) to address the challenges of enhancing performance accuracy and generalizability for McSL recognition. GmTC is designed to outperform high-performance convolutional models and canonical transformers. Unlike many previous transformer-based hand gesture recognition systems that segmented the input image into patches and extracted features individually, resulting in poorly constructed models and the implementation of linear projections, GmTC takes a different approach. The proposed GmTC system constructs a hybrid network by leveraging the superpixel-based GCN for local features and the long-range dependency of features from MHSA with CNN. This innovative design enhances the model's effectiveness by considering spatial distance-based relationships among super-pixels.

To do this, we employed two parallel streams: the superpixel-based GCN and general deep learning streams. In the GCN stream, superpixels were initially computed using the SLIC approach. These superpixels were then treated as nodes in a fully connected graph, enabling the extraction of spatial relationships among them to derive effective features. This stream specifically utilized a GCN to calculate distance-based super-pixel relationship features. In the second stream, self-attention-based features were extracted. This involved passing the features through multiple stages of the MHSA and CNN modules, inspired by existing architectures such as CMT [20], ResNet-50 [40], and DeiT [41]. The attention-based general deep learning stream addresses fixed-size patch issues and extracts multiscale features using a grain module. The output undergoes four stages of the MHSA and CNN module, employing multiple multi-head

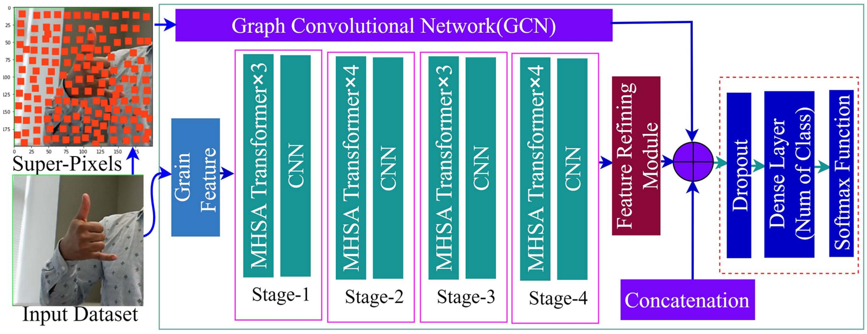

FIGURE 4. Architecture of the proposed model.

self-attention transformers (MHSAT) blocks sequentially in each stage. Extracted features are stacked to maintain input resolution. A feature refining module enhances and selects potential features. The GCN feature is concatenated with the general deep learning feature, creating the final feature. The process concludes with a classification module containing a fully connected layer and a softmax-based n-way classification layer.

A. SLIC-BASED GCN STREAM

Superpixel-based graph convolutional methods represent a transformative approach in computer vision, delivering improved computational efficiency, noise resilience, and semantic information extraction [45]. These techniques group pixels into semantically meaningful superpixels, optimizing graph convolutions for more efficient processing of larger images. Constructing a graph by considering all image pixels leads to high computational complexity. To address this, we propose leveraging superpixel-based node graphs instead of pixel-based nodes to exploit the capabilities of graph neural networks fully [31], [46].

1) SIMPLE LINEAR ITERATIVE CLUSTERING (SLIC)

In pursuit of efficient McSL recognition, our approach involves leveraging SLIC-based superpixels, which can significantly reduce the size of graph nodes compared to traditional image pixels. Superpixels, compact image segments defined by shared characteristics such as color and location, offer an advantageous intermediate representation. Notably, the SLIC algorithm, chosen for its stability and rapid segmentation speed, is employed in our work for image superpixel segmentation aiming for computational efficiency by eliminating redundant pixel values, making it particularly conducive to learning models by reducing learnable parameters [27], [47], [48]. We specifically adopted the SLIC approach to partition gesture images into spatially connected superpixels [36]. This strategy allowed us to explore the spatial structure of gesture images more effectively. In our implementation, the number of superpixels (N) is set equal to the image's width, providing optimal coverage and ensuring comprehensive spatial analysis [31]. This innovative application of superpixels in hand gesture recognition contributes to improved computational efficiency and more nuanced spatial understanding, marking our research's novel and impactful aspect. The SLIC algorithm for the hand gesture image undergoes a dimension reduction, resulting in an $N \times N$ dimension superpixel representation shown in Fig. 4 as superpixels. This captured the spatial intricacies of the gesture, providing a compact and informative representation that serves as a foundation for subsequent stages in our hand gesture recognition process. This dimensionality reduction contributes to improved computational efficiency. It allows for a more focused analysis of relevant spatial features, highlighting the effectiveness of the SLIC-based superpixel approach in our research.

2) GRAPH CONVOLUTIONAL NETWORK (GCN)

We implement an adjacency relation among the superpixels of the SL images that can be represented as an undirected graph using the following (1).

$$G = (V, E) \tag{1}$$

Here, V is the set of vertices, and E is the edge. Practically, vertex and node can be encoded into the node matrix and an adjacency matrix. After that, we applied a graph convolutional neural network. In conventional neural networks, linear layers use a linear transformation of the input data. This transformation involves converting input x superpixel features into the hidden feature H using the following (2).

$$H = Wx + b \tag{2}$$

where b denotes the biased data, in the graph structure superpixel data, we must add an extra connection among the superpixels. In addition, we considered it as a graph; an adjacency matrix is inevitable here and that $x = (x \cdot A)$. We can enhance the representation of a node by combining its features with those of its neighbours using convolution or neighbourhood aggregation using below (3):

$$H = \sum_{i=1}^{N} W_i.(x.A) + b_i \tag{3}$$

In this case, the weight matrix W is unique and sharable with all neighbour nodes. In addition, every node does not have an equal number of neighbours, and it may be different. In this situation, normalization using the degree of node D can be a solution to ensure a similar range of values for each node [45].

$$H = \frac{1}{D}\sum_{i=1}^{N} W_i.(x.A) + b_i \tag{4}$$

Now, from (4), we can rewrite the graph-based neural network model that we used in our study. We considered the multi-layer graph convolutional neural network (GCN) for the superpixel-based node using the layer-wise propagation rules in the equation described below. The partial derivative symbol can be represented as (5).

$$GNN(x, A) = \sigma\left(\sum_{l=1}^{L} W_l.\left(x.\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\right)W_l + b_l\right) \tag{5}$$

where L represents the number of layers of the graph neural network, the adjacency matrix of the self-connection in the undirected graph is represented by $\tilde{A} = A + I_N$ and I_N denotes the identity matrix. Layer specific trainable weight matrix represented by W_l and $\tilde{D}_{ij} = \sum_{l=1}^{L}\sum_l \tilde{A}_{ij}$. Activation function denoted by $\sigma(.)$ and in our cases we used $ReLU(.) = \max(0, .)$

The final feature generated by the GCN stream can be defined by the following (6).

$$Graph_{\text{Feature}} = \text{avg}\left(\underbrace{\text{GCN}_L \circ \ldots \circ \text{GCN}_2 \circ \text{GCN}_1}_{L\text{ layers}}\right) \tag{6}$$

where L denotes the number of layers and Graph convolution is used here as the number of layers, ∘ denotes the composition of functions. The composition (∘) is used to indicate the sequential application of the GraphConvolution functions. Graph convolutions on superpixels of the hand gesture image are effective in capturing relationships compared to convolutions applied directly to pixels. This approach enhances the extraction of effective features for hand gesture recognition, with the resulting set of features denoted as $Graph_{\text{Feature}}$, serving as the final feature representation of the first stream. The utilization of superpixels optimizes the modelling of spatial dependencies within the SL, and this innovative strategy aligns with the overarching goal of improving the efficiency and accuracy of hand gesture recognition systems.

B. SPATIAL ATTENTION BASED GENERAL DEEP LEARNING STREAM

The second stream is composed of a grain module, an MHSA Transformer, MLP Convolution and a feature refining module. The concept of this stream developed from the architecture Shin [21], CMT [20], ResNet-50 [30], and DeiT [32]. We can define the grain feature extractor model as $G_{\text{Feature}}(\theta_G, X_R^i)$ where θ_{Grain} denoted the weight of the model. Here, the input data dimension is represented with $X \in R^{H\times W\times d}$ and height, width and channel are represented with H, W and d, respectively. The details of this branch are described below.

1) GRAIN MODULE

The grain module inputted the original image, generating the fine-grained initial feature extraction. We developed this model by following the ResNet [20], [49] technique. We divide the module into two stages. Where the first block of the module consists of a $three \times three$ two convolutional layer with stride two and a second block, we use a three × three one convolutional layer with stride one and produce 32 output channels aiming to reduce the input size. Where the second stage of the module is mainly used to perform the patch aggregation method, including the convolutional layer and normalization layer, we can express the grain feature extractor model as $G_{\text{Feature}}(\theta_G, X_R^i)$ where θ_G denotes the weight of the model. Here, the input data dimension is represented with $X \in R^{H\times W\times d}$ and height, width and channel are represented with $H, Wandd$, respectively. Fig. 5(a) demonstrated the grain architecture.

2) INITIALIZATION MODULE

In the MHSA Transformer, we first employed an initial model for extracting local features from the grain feature as position encoding techniques [20]. The main purpose of this module is to discuss different augmentations such as shift and rotation, which two are considered the most important manners in the visual task, and it is not good to avoid this operation. In addition, this module also helps us to overcome the image translation dependency on the system [37], [50], [51]. Our initial module can solve the local relation-related problems, which can be extracted using the following (7).

$$IM(X_R^i) = EWConv(\theta_{IM}, G_R^i) + G_R^i \tag{7}$$

Here, the initial module feature is contained in the IM variables, an element-wise convolutional operation denoted by $EWConv$. Moreover, $G \in R^{(H\times W\times d)}$ represented the feature of the grain module, and height, width, and channel represent H, W and d, respectively.

3) MULTIHEAD SELF ATTENTION (MHSA)

Multihead self-attention (MHSA) [14], [40] has recently proven to be excellent in both computer vision and NLP-related research. Fig. 6 demonstrates the MHSA architecture. The main concept of the self-attention model is to include a query, key, and value matrix. Firstly, self-attention can take input in the following format: $X \in \mathbb{R}^{n\times d}$. It is then transformed into the mentioned three matrices, defined as $Q \in \mathbb{R}^{n\times d}$, $K \in \mathbb{R}^{n\times d}$, and $V \in \mathbb{R}^{n\times d}$, respectively. In this study, the number of patches is represented as $n = H \times W$, where H and W denote height and width. The data dimensions for the three matrices are denoted as d, d_k, and d_v for the query, key, and value, respectively. The MHSA can be expressed by the following (8).

$$SA = softmax\left(\frac{qk^T}{\sqrt{d_k}}\right) \times v \tag{8}$$

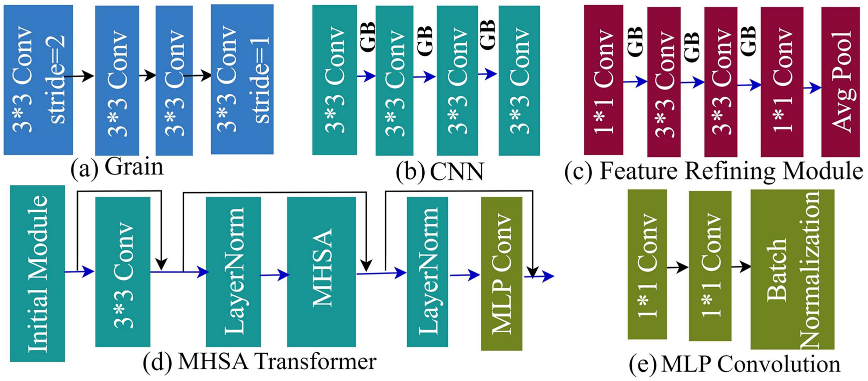

FIGURE 5. (a) Grain model. (b) CNN feature extraction module. (c) feature refining module. (d) MHSA transformer. (e) MLP convolution.

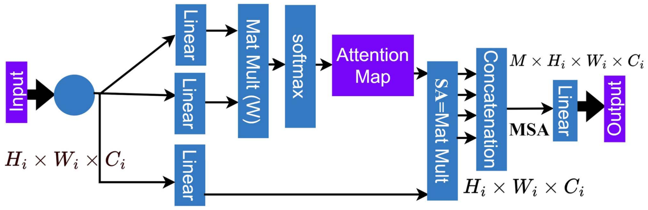

FIGURE 6. Architecture of MHSA attention module.

where SAT is the self-attention, query, key, value and dimension represented by q, k, v and d_k, respectively, the mechanism of the attention module is to multiply the query key matrix, then use an activation layer and make an attention map. Afterwards, we performed matrix multiplication between the value and attention map and generated output for the single head and four head in our cases. We repeated the same procedure four times and concatenated the four head values to produce the final MHSA feature. Finally, we applied the activation layer and produced the MHSA feature to feed the MLP convolution or next module. We can write the Initial model and Multihead attention output below the equation, which takes the output of the Grain feature as input.

$$\overline{G_R^i} = IM(G_R^{(i-1)}) \tag{9}$$

$$\overline{\overline{G_R^i}} = MHSA(LN(\overline{G_R^i}) + \overline{G_R^i}) \tag{10}$$

Here, $\overline{G_R^i}$ and $\overline{\overline{G_R^i}}$ denotes the initial module (IM) and MHSA module feature for the individual stage I consequently where layer normalization denoted with LN. Finally, we can write the final feature of the MHSA as $MHSA_R^i$, which we sent to the convolutional layer.

4) CNN MODULE

Fig. 5(b) demonstrated the CNN architecture which we used here after the MHSA module for further enhancement. This module aims to incorporate spatial information from the local region of output MHSA [52]. Here, we develop the CNN model by including four convolutional layers that are incorporated with GeLU activation and batch normalization.

5) MLP CONVOLUTION

We employed the multilayer perception convolution block after the attention in the MHSA. In the MLP block, we included a single block of the 1×1 two convolution layers [53]. We used the GeLU activation function and normalization layer after the first one-by-one convolution layer and a batch normalization layer. In the same for the 2nd convolutional layer seems to be a general convolutional layer, but we used kernel size 1, aiming to work for 1 pixel for the input image. The main purpose of using CNN is to extract two-dimensional neighbourhood structures, whereas MLPConv, after MHSA, converted the global MHSA into local pixel information. The output of the MLP convolution can be defined by the following (11). Fig. 5(e) demonstrated the MLP convolution architecture.

$$MLP_R^i = \overline{\overline{G_R^i}} = MLPConv(conv(\overline{\overline{G_R^i}})) \tag{11}$$

C. FEATURE REFINING MODULE

After generating the feature from the multi-stage of the CNN and MHSA transformer, we used a feature refining module to refine the feature, aiming to detect effective features to improve the performance, accuracy and efficiency of the systems. To implement this module, we follow the FFN for the ViT transformer [44], where the demonstrated the two linear layers, which are separated using a GeLU activation function [16], [19]. Fig. 5(d) demonstrated the schematic diagram of the feature refining module, which is also made by following the inverted residual feed-forward network (IRFFN) [20]. The output of this module can be defined as the following (12).

$$RF_{MLP} = Avg(EWconv(conv(conv(EWconv(MLP_R^i))))) \tag{12}$$

where RF_{MLP} represents the output of the feature refining module, in addition, Avg defined the averaging pooling layer. We included the GeLU activation function and Batch Normalization in each layer. The elementwise convolutional neural network calculates local information with a minimum cost and value. Then, we employed a global average pooling layer to produce the matrix's feature vector by averaging the samplewise features.

TABLE 2. Possible Hyperparameters

Hyperparameter Name	Proposed Model	Existing Transfer Learning
Training : Testing	Dataset Protocol	70%:30%
Dp rate	0.01	0.01
Learning Rate	5e-6 to 1e-3	1e-3
Optimizer	Adam	Adam
Batch Size	8 - 32	8
Epochs	50	50-500

D. FEATURE CONCATENATION AND CLASSIFICATION MODULE

After extracting the GCN and deep learning-based MHSAT features, we concatenated them to produce the final features according to (13), which we fed the classification module with. The output of the concatenation is shown in (13). After that, we applied the dropout layer on the concatenated feature, and the dropout rate followed the 40% rules. Then, we set a density layer as an activation function for the softmax activation. Finally, this softmax action function produces a probabilistic confidence map, which we considered a feature map. This probabilistic confidence map is generated as a number of features equal to the associated dataset classes.

$$Final_{\text{Feature}} = Graph_{\text{Feature}} \bigoplus RF_{MLP} \tag{13}$$

V. EXPERIMENTAL EVALUATION

We conducted various experiments to evaluate the proposed system's superiority, effectiveness and generalizability, including diverse language datasets to build the McSL recognition system.

A. TRAINING SETTING

Table 1 demonstrated the dataset information used in the study to evaluate the proposed model. We used four multi-culture SL datasets: Japanese, Korean, Bangla and ASL. To divide the dataset into the training, we follow the state-of-the-art model strategy, and in most cases, it is 70% as a training dataset and 30% as a testing dataset. Table 2 showed the various hyperparameter ranges which we used in the study. In our study, our architecture was instantiated within the PyTorch framework on NVIDIA 8 GB GPU machines. For the compilation phase, we opted for the Adam optimizer as the optimization method, employing a learning rate of 0.001 during the model training. The batch size was configured to 32, and a dropout rate of 0.2 was applied.

B. ABLATION STUDY

Our model consists of a superpixel-based GCN module and a CNN, MHSA-based general deep learning branch. The GCN incorporates multiple layers for effectiveness, utilizing a superpixel-based graph structure. The general deep learning module comprises multi-stages of CNN and MHSA, with four stages in our study. The performance analysis in the table below covers the McSL model on diverse datasets and branches. According to the Table 3, we can say that two-stream fusion features can improve the performance accuracy in this strategy.

TABLE 3. Strategic Ablation Study Highlighting Variations in GCN and General CNN Branch

Dataset Name	Learning Rate and Optimizer	GCN Branch	General CNN and MHSA Attention Branch	Combined Branch
KSL-20	0.003, Adam	88.00	95.00	100.00
KSL-77	0.003, Adam	80.00	92.00	99.30

TABLE 4. Performance Result of the KSL Datasets and State-of-the-Art Comparison

Method Name	KSL-77 dataset (%)	KSL-20 dataset
Vision Transformer [23]	88.00	93.00
ResNet	88.00	95.00
InceptionResnet	88.00	95.00
DenseNet	88.00	95.00
TSN [22]	79.80	Na/
Shin [21]	89.00	98.00
Proposed Model	99.33	100.00

C. PERFORMANCE WITH THE KSL DATASET

Table 4 demonstrated the performance accuracy of the proposed model with KSL-77 and the KSL-20 datasets, where our proposed model achieved 99.33% and 100.00% accuracy, respectively. The table also reported performance accuracy with transfer learning and state-of-the-art model performance. Yang et al. applied a CNN mode where they reported 79.00% accuracy [22]. Shin et al. applied parallel of the CNN and attention model and achieved 89.00% and 98.00% accuracy for the KSL-77 and lab KSL dataset, respectively [21]. Fig. 7 demonstrated the accuracy and loss curve for the KSL-77 dataset. The KSL 77 is a large dataset that contains 77 class labels, and our proposed model achieved high-performance accuracy compared to the state-of-the-art model.

D. PERFORMANCE WITH THE ASL DATASET

We also assessed our model using two ASL datasets, ASL-10 and ASL-20, employing various transfer learning techniques. Table 5, showcase our model's strong performance, achieving 99.46% and 99.60% accuracy for ASL-10 and ASL-20 datasets, respectively. Rahim et al. applied CNN and SVM for feature extraction and classification, reporting 97.00% accuracy for our lab ASL dataset [28]. Miah et al. also employed advanced augmentation and segmentation techniques, achieving 99.30% accuracy with our lab ASL dataset [9]. In summary, our proposed model demonstrates superior accuracy compared to existing models. Notably, these accuracy rates surpass those reported for transfer learning and existing of the art model mentioned in the table.

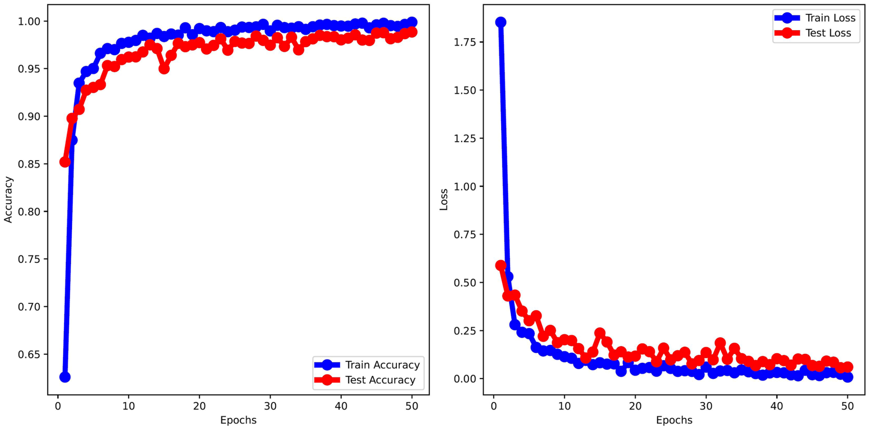

FIGURE 7. Accuracy and loss curve for the KSL-77 dataset.

TABLE 5. Performance Result of the ASL Datasets and State-of-the-Art Comparison

Model Name	ASL KLP Dataset [%]	ASL-20 Wrod Dataset [%]
VGG19	96.30	97.30
ResNet	96.30	96.90
InceptionResnet	99.46	99.46
DenseNet	96.30	97.30
SVM [8]	89.70	n/a
SVM [13]	96.30	n/a
Rahim [28]	n/a	97.00
Miah [9]	n/a	99.30
Proposed Model	99.46	99.60

TABLE 6. Performance Result of the BSL Datasets and State-of-the-Art Comparison

Model Name	Existing BSL Dataset [%]	Our Lab BSL Dataset [%]
VGG19 [29]	89.60	92.00
Concatenate CNN [30]	91.52	n/a
BenSignNet [10]	93.00	n/a
ResNet	89.60	92.00
InceptionResnet	89.60	92.00
DenseNet	89.60	92.00
Proposed Model	93.50	96.88

TABLE 7. Performance Accuracy of the Proposed Model With JSL and Digit Datasets

Method	NTU [2] (%)	Senz3D[4] (%)	JSL[%]
Zhou Ren [1]	93.20	Na	-
PG2 [24]	93.66	98.73	-
Yan et al. [25]	95.33	99.49	-
Ma et al. [24]	95.86	99.05	-
PoseGAN [26]	96.12	99.54	-
GestureGan [25]	96.66	99.74	-
VGG19	96.66	99.74	-
ResNet	96.66	99.74	89.00
InceptionResnet	96.66	99.74	88.00
DenseNet	96.66	99.74	90.00
GoogleNet [7]	na	na	90.00
Proposed Model	97.22	99.74	92.37

E. PERFORMANCE WITH THE BSL DATASET

We demonstrated the performance accuracy of the proposed model with the BSL dataset. Table 6 demonstrated the performance accuracy where our proposed model achieved 93.50% accuracy for the existing BSL dataset and 96.88% accuracy achieved for the lab dataset. The table also reported the performance accuracy with the transfer learning and the state-of-the-art comparison table. Rafi et el. Employed the VGG19 method for the BSL recognition, where they achieved 89.60% accuracy, and we reported 92.00% accuracy for the lab dataset [29]. Abedin et al. used a concatenated CNN model, which generated 91.52% accuracy for the existing BSL dataset [30]. In addition, Musa et al. employed the BenSingNet model on the existing BSL dataset, which generated 93.00% accuracy [10]. Based on the performance accuracy in the table, our performance model achieved higher performance accuracy than the existing model.

F. PERFORMANCE WITH THE JSL, DIGIT AND NTU DATASETS

Table 7 demonstrates the performance accuracy of the proposed model with the JSL dataset for the proposed model, where it reported that the proposed model achieved 92.37% accuracy. It also showed the performance for NTU and Senz3D datasets, where our model achieved 97.22% and 99.74% performance accuracy, respectively. Then, we reported performance for the state-of-the-art models, including PoseGAN [26], GestureGAN [25], and Ma et [24] and several transformer learning. Among them, Ren et al. developed a hand gesture recognition based on a template-matching method where they reported 93.00% accuracy [54]. The pose Guided Person Generation Network (PG2) method is reported 93.66% and 98.73% accuracy for the NTU and Senz3D

TABLE 8. State of the Art Comparison With LSA64 Dataset

Method Name	Number of class	Performance Accuracy [%]
3D CNN [55]	64	93.90
Cumulative shape difference + SVM [%] [56]	64	85.00
Inception CNN + LSTM [57]	64	95.20
Inception CNN + BiLSTM [59]	64	96.00
3D CNN + LSTM [58]	40	98.50
Proposed Model	64	99.10

datasets, respectively. In the same way, the author in [25] reported 95.33% and 99.49% accuracy. Siarohin et al. reported 96.12% and 99.54% accuracy for NTU and Senz3D datasets, respectively [26]. Tang et al. proposed Gesture GAN methodology including a generator and discriminator, by taking RGB image input for the generation and classification and achieved 96.66% and 99.74% for the NTU and Senz3D datasets, respectively [25].

G. STATE OF THE ART-COMPARISON WITH A BENCHMARK LSA64 DATASET

Table 8 demonstrated the state-of-the-art comparison of the proposed model with the LSA64 dataset. The 3D CNN [55] method, employing a three-dimensional convolutional neural network, achieves an accuracy of 93.90% for 64 classes. In contrast, the cumulative Shape Difference + SVM approach [56], combining shape differences cumulatively with a support vector machine, yields an accuracy of 85.00%. The Inception CNN + LSTM [57] and Inception CNN + BiLSTM models [57], incorporating long short-term memory networks, demonstrate higher accuracies at 95.20% and 96.00%, respectively, for 64 classes. A notable configuration is the 3D CNN + LSTM [58] method, achieving an impressive accuracy of 98.50% for 40 classes. Our proposed model achieves high-performance accuracy with 64 sign words, proving its superiority compared to the state-of-the-art model evaluated in Table 8.

VI. DISCUSSION

In the study, we proposed a hybrid GmTC model to enhance effective features of hand gesture recognition, aiming to produce good performance accuracy compared to the existing state-of-the-art model. Besides the performance accuracy, we also tried to stabilize the computational complexity of the proposed model with the recently developed system. We calculated the parameter and computational complexity of the proposed system. Notably, the consistent parameters of the proposed model are 8 million across the dataset for 32 × 32 pixel image, and the computation complexity value of 130 BFLOP for each batch across diverse datasets indicates the proposed model's uniform demand for processing resources. This insight proves invaluable for assessing the model's efficiency and scalability, irrespective of dataset variations. Our ultimate goal is to contribute to the establishment of a Multicultural Sign Language (McSL) communication system, particularly benefiting the hard of hearing and mute-hearing community. To achieve this, the proposed system is seamlessly deployable for real-time applications. Leveraging a pre-trained GmTC model saved as a pickle file, a user interface (UI) tailored for desktop, web, or mobile applications integrates a menubar, buttons, and input/output boxes. The UI must include real-time gesture capture; the GmTC model processes the input sign language gesture, providing an immediate and dynamic response. The output box displays predicted values, creating an interactive and engaging user experience. This deployment strategy for McSL recognition supports the hard of hearing and mute community, empowering researchers to integrate our system effortlessly.

VII. CONCLUSION

In our study, we proposed GmTC, a novel model for McSL recognition, by integrating graphs and general DNN. The proposed model is constructed with two streams. The GmTC system synergistically utilizes GCN, local CNN features, and long-range dependencies from multi-head self-attention, compelling the model to attain diverse discriminative features such as short-range, long-range, and graph-based extractions. Our primary objective was to extract extensive distance-based pixel relationships, demonstrating the efficacy of GCN in image-based tasks. Consequently, the GmTC model learns these adaptive features, enhancing generalization capabilities. The proposed method achieved its goal by producing high-performance accuracy with diverse SLR datasets (JSL, KSL, BSL, ASL, and LSA64). The outcomes revealed consistently high-performance accuracy, affirming the effectiveness and generalizability of our approach. The comprehensive evaluation showcased the model's superiority over high-performance CNN and canonical transformer models. In the future, we aim to deploy this model as a streamlined, generalized McSL system by including ten SLs and optimizing parameters for enhanced speed in multimodal applications.

REFERENCES

[1] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Trans. Multimedia*, vol. 15, pp. 1110–1120, 2013.

[2] A.S.M. Miah, J. Shin, M.A.M. Hasan, Y. Okuyama, and A. Nobuyoshi, "Dynamic hand gesture recognition using effective feature extraction and attention based deep neural network," *Proc. IEEE 16th Int. Sympos. Embedded Multicore/Many-core Systems-on-Chip*, 2023, pp. 241–247.

[3] A. S. M. Miah, M. A. M. Hasan, J. Shin, Y. Okuyama, and Y. Tomioka, "Multistage spatial attention-based neural network for hand gesture recognition," *Computers*, vol. 12, no. 1, 2023, Art. no. 13.

[4] J. Murray, K. Snoddon, M. Meulder, and K. Underwood, "Intersectional inclusion for deaf learners: Moving beyond general comment no. 4 on article 24 of the united nations convention on the rights of persons with disabilities," *Int. J. Inclusive Educ.*, vol. 24, pp. 691–705, 2018.

[5] I. Roux et al., "CHD7 variants associated with hearing loss and enlargement of the vestibular aqueduct," *Hum. Genet.*, vol. 142, pp. 1499–1517, 2023.

[6] J. Lee, C. Y. Yoon, J. Lee, T. H. Kong, and Y. J. Seo, “A situational analysis of ear and hearing care in South Korea using who ear and hearing care situation analysis tool,” *Front. Public Health*, vol. 11, 2023, Art. no. 1215556.

[7] J. Shin, M. A. M. Hasan, A. S. M. Miah, K. Suzuki, and K. Hirooka, “Japanese sign language recognition by combining joint skeleton-based handcrafted and pixel-based deep learning features with machine learning classification,” *Comput. Model. Eng. Sci.*, pp. 1–21, 2024. [Online]. Available: https://www.techscience.com/CMES/online/detail/19824

[8] G. Dewaele, F. Devernay, and R. Horaud, “Hand motion from 3D point trajectories and a smooth surface model,” in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 495–507.

[9] S. J. Miah, A. S. Musa, M. A. M. Hasan, M. A. Rahim, and Y. Okuyama, “Rotation, translation and scale invariant sign word recognition using deep learning,” *Comput. Syst. Sci. Eng.*, vol. 44, no. 3, pp. 2521–2536, 2023.

[10] A. S. M. Miah, J. Shin, M. A. M. Hasan, and M. A. Rahim, “Bensignnet: Bengali sign language alphabet recognition using concatenated segmentation and convolutional neural network,” *Appl. Sci.*, vol. 12, no. 8, 2022, Art. no. 3933.

[11] M. F. Nurnoby and E.-S. M. El-Alfy, “Multi-culture sign language detection and recognition using fine-tuned convolutional neural network,” in *Proc. Int. Conf. Smart Comput. Appl.*, pp. 1–6, 2023.

[12] G. Marin, F. Dominio, and P. Zanuttigh, “Hand gesture recognition with jointly calibrated leap motion and depth sensor,” *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 14991–15015, 2016.

[13] S. H. I. Yuanyuan, L. I. Yunan, F. U. Xiaolong, M. Kaibin, and M. Qiguang, “Review of dynamic gesture recognition,” *Virtual Reality Intell. Hardware*, vol. 3, no. 3, pp. 183–206, 2021.

[14] A. S. M. Miah, M. A. M. Hasan, and J. Shin, “Dynamic hand gesture recognition using multi-branch attention based graph and general deep learning model,” *IEEE Access*, vol. 11, pp. 4703–4716, 2023.

[15] S. Jetley, N. A. Lord, N. Lee, and P. H. S. Torr, “Learn to pay attention,” in *Proc. Int. Conf. Learn. Representations*, Vancouver, BC, Canada, 2018, pp. 1–10.

[16] Y. Iwai, K. Watanabe, Y. Yagi, and M. Yachida, “Gesture recognition by using colored gloves,” in *Proc. IEEE Int. Conf. Syst., Man Cybern. Inf. Intell. Syst.*, 1996, pp. 76–81.

[17] A. D. Wilson and A. F. Bobick, “Parametric hidden Markov models for gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 884–900, Sep. 1999.

[18] H.-K. Lee and J.-H. Kim, “An HMM-based threshold model approach for gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 961–973, Oct. 1999.

[19] C. Kwok, D. Fox, and M. Meila, “Real-time particle filters,” *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 15, 2002.

[20] J. Guo et al., “CMT: Convolutional neural networks meet vision transformers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12165–12175.

[21] J. Shin et al., “Korean sign language recognition using transformer-based deep neural network,” *Appl. Sci.*, vol. 13, no. 5, 2023, Art. no. 3029.

[22] S. Yang, S. Jung, H. Kang, and C. Kim, “The Korean sign language dataset for action recognition,” in *Proc. Int. Conf. Multimedia Model.*, 2020, pp. 532–542.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[24] L. Ma, Q. Sun, S. Georgoulis, L. V. Gool, B. Schiele, and M. Fritz, “Disentangled person image generation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 99–108.

[25] H. Tang, W. Wang, D. Xu, Y. Yan, and N. Sebe, “Gesturegan for hand gesture-to-gesture translation in the wild,” in *Proc. 26th ACM Int. Conf. Multimedia*, 2018, pp. 774–782.

[26] A. Siarohin, E. Sangineto, S. Lathuiliere, and N. Sebe, “Deformable GANs for pose-based human image generation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3408–3416.

[27] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, pp. 167–181, 2004.

[28] M. A. Rahim, M. R. Islam, and J. Shin, “Non-touch sign word recognition based on dynamic hand gesture using hybrid segmentation and CNN feature fusion,” *Appl. Sci.*, vol. 9, no. 18, 2019, Art. no. 3790.

[29] A. M. Rafi, N. Nawal, N. S. N. Bayev, L. Nima, C. Shahnaz, and S. A. Fattah, “Image-based Bengali sign language alphabet recognition for deaf and dumb community,” in *Proc. IEEE Glob. Humanitarian Technol. Conf.*, 2019, pp. 1–7.

[30] T. Abedin, K. S. S. Prottoy, A. Moshruba, and S. B. Hakim, “Bangla sign language recognition using a concatenated BDSL network,” in *Computer Vision and Image Analysis for Industry 4.0*. Boca Raton, FL, USA: Chapman and Hall/CRC, 2023, pp. 76–86.

[31] J.-H. Bae et al., “Superpixel image classification with graph convolutional neural networks based on learnable positional embedding,” *Appl. Sci.*, vol. 12, no. 18, 2022, Art. no. 9176.

[32] W. Tao, M. C. Leu, and Z. Yin, “American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion,” *Eng. Appl. Artif. Intell.*, vol. 76, pp. 202–213, 2018.

[33] E.-J. Ong, H. Cooper, N. Pugeault, and R. Bowden, “Sign language recognition using sequential pattern trees,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2200–2207.

[34] G. Yuan, X. Liu, Q. Yan, S. Qiao, Z. Wang, and L. Yuan, “Hand gesture recognition using deep feature fusion network based on wearable sensors,” *IEEE Sensors J.*, vol. 21, no. 1, pp. 539–547, Jan. 2021.

[35] S. Aly and W. Aly, “DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition,” *IEEE Access*, vol. 8, pp. 83199–83212, 2020.

[36] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. 3rd Int. Conf. Learn. Representations*, California, USA, 2015, pp. 1–14.

[37] A. A. Barbhuiya, R. K. Karsh, and S. Dutta, “Alexnet-CNN based feature extraction and classification of multiclass ASL hand gestures,” in *Proc. 5th Int. Conf. Microelectronics, Comput. Commun. Syst.*, 2021, pp. 77–89.

[38] C. Szegedy et al., “Going deeper with convolutions,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[39] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[40] A. Vaswani et al., “Attention is all you need,” *Proc. Annu. Conf. Neural Inf. Process. Syst*, vol. 30, 2017.

[41] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020, *arXiv:2010.11929*.

[42] K. H. Tarafder, N. Akhtar, M. M. Zaman, M. A. Rasel, M. R. Bhuiyan, and P. G. Datta, “Disabling hearing impairment in the Bangladeshi population,” *J. Laryngol. Otol.*, vol. 129, no. 2, pp. 126–135, 2015.

[43] F. Ronchetti, F. Quiroga, C. Estrebou, L. Lanzarini, and A. Rosete, “LSA64: A dataset of Argentinian sign language,” in *Red de Universidades con Carreras en Informática*, pp. 794–803, 2016. [Online]. Available: http://sedici.unlp.edu.ar/handle/10915/56764

[44] X. Chu, Z. Tian, B. Zhang, X. Wang, and C. Shen, “Conditional positional encodings for vision transformers,” in *Proc. 11th Int. Conf. Learn. Representations*, Kigali Rwanda, 2023.

[45] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” Toulon, France, 2017.

[46] Q. Liu, L. Xiao, J. Yang, and Z. Wei, “CNN-enhanced graph convolutional network with pixel- and superpixel-level feature fusion for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8657–8671, Oct. 2021.

[47] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[48] A. Vedaldi and S. Soatto, “Quick shift and kernel methods for mode seeking,” in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 705–718.

[49] F. Wang et al., “Residual attention network for image classification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6450–6458.

[50] A. Howard et al., “Searching for MobileNetV3,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.

[51] A. Jalal, S. Kim, and B. J. Yun, “Assembled algorithm in the real-time H.263 codec for advanced performance,” in *Proc. 7th Int. Workshop Enterprise Netw. Comput. Healthcare Ind.*, 2005, pp. 295–298.

[52] R. Rastgoo, K. Kiani, and S. Escalera, “Hand sign language recognition using multi-view hand skeleton,” *Expert Syst. Appl.*, vol. 150, 2020, Art. no. 113336.

[53] W. Wang et al., “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 548–558.

[54] M. Panwar, “Hand gesture recognition based on shape parameters,” in *Proc. Int. Conf. Comput., Commun. Appl.*, 2012, pp. 1–6.

[55] G. M. R. Neto, G. B. Junior, J. D. S. d. Almeida, and A. C. d. Paiva, “Sign language recognition based on 3D convolutional neural networks,” in *Proc. Int. Conf. Image Anal. Recognit.*, 2018, pp. 399–407.

[56] J. Rodríguez and F. Martínez, “Towards on-line sign language recognition using cumulative SD-VLAD descriptors,” in *Proc. Colombian Conf. Comput.*, 2018, pp. 371–385.

[57] S. Masood, A. Srivastava, H. C. Thuwal, and M. Ahmad, “Real-time sign language gesture (word) recognition from video sequences using CNN and RNN,” in *Proc. Intell. Eng. Inform.*, 2018, pp. 623–632.

[58] E. K. Elsayed and D. R. Fathy, “Semantic deep learning to translate dynamic sign language,” *Int. J. Intell. Eng. Syst.*, vol. 14, no. 1, 2021.

[59] J. A. Shah et al., “Deepsign: A deep-learning architecture for sign language,” M.S. dissertation, The Univ. Texas Arlington, Nedderman Drive Arlington, Texas, USA, 2018.

ABU SALEH MUSA MIAH received the B.Sc. Engg. and M.Sc.Engg. degrees in computer science and engineering from the Department of Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh. He has been working toward the Ph.D. degree with the School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan, since 2021, under a scholarship from the Japanese government (MEXT). His research interests include computer vision and deep learning.

MD. AL MEHEDI HASAN (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science and engineering from the Department of Computer Science and Engineering, University of Rajshahi, Rajshahi, Bangladesh. His research interests include bioinformatics, artificial intelligence, pattern recognition, medical images, signal processing, machine learning, computer vision, data mining, and big data analysis.

YOICHI TOMIOKA (Member, IEEE) received the B.E., M.E., and D.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan. He was a Research Associate with the Tokyo Institute of Technology, Tokyo, Japan, till 2009. He was an Assistant Professor with the Division of Advanced Electrical and Electronics Engineering, Tokyo University of Agriculture and Technology till 2015. He has been a Senior Associate Professor with the University of Aizu, Aizuwakamatsu, Japan, since 2019. His research interests include image processing, hardware acceleration, high-performance computing, electrical design automation, and combination algorithms.

JUNGPIL SHIN (Senior Member, IEEE) received the B.Sc. degree in computer science and statistics and the M.Sc. degree in computer science from Pusan National University, Busan, South Korea, and the Ph.D. degree in computer science and communication engineering from Kyushu University, Japan, under (MEXT). He was an Associate Professor, a Senior Associate Professor, and a Full Professor with the School of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan. His research interests include pattern recognition, image processing, and computer vision. He is a member of ACM, IEICE, IPSJ, KISS, and KIPS. He was the Program Chair and as a program committee member for numerous international conferences. He is the Editor of IEEE journals Springer, Sage, Taylor and Francis, MDPI Sensors and Electronics, and Tech Science.