Patient Assistance System Based on Hand Gesture Recognition

H. Pallab Jyoti Dutta, M. K. Bhuyan, *Senior Member, IEEE*, Debanga Raj Neog, Karl Fredric MacDorman, and Rabul Hussain Laskar

***Abstract*— We propose a two-stage hand gesture recognition architecture to support a patient assistance system. Some medical conditions limit mobility, and the patient must rely on medical staff to meet their needs. In such cases, a phone or intercom is not convenient to call for help. A vision-based system operated by changing the orientation of fingers can be used to send specific messages without making arm movements. However, vision-based hand gesture recognition is hindered by occlusion, background clutter, and variations in illumination. Therefore, we developed a two-stage architecture: the first stage produces a saliency map to simplify recognition and the second stage performs classification. A novel combined loss function optimizes the saliency detection model and makes the saliency map more precise. An adaptive kernel-based channel attention layer is used to emphasize salient features. The proposed architecture achieved precise saliency detection on four benchmark datasets and high-accuracy recognition on two. We designed an interface for patients to send specific messages to the medical staff using hand gestures. The interface help patients request assistance and connect with medical staff without leaving the bed or involving a third party.**

***Index Terms*— Channel attention, convolution neural network (CNN)–transformer network, hand gesture recognition, patient assistance, saliency detection, virtual interface.**

I. Introduction

HAND gestures offer a natural and spontaneous way to communicate [1], [2], [3]. People convey much information through hand gestures, especially those with speech disorders, disabilities, or in hospital. Thus, hand gesture recognition plays a vital role in communication. This article explores the possibility of developing an assistive system for smooth interaction between patients and medical staff via simple hand gestures.

Whenever a patient requires assistance, the medical staff should attend to their requirements swiftly and effectively. The communication between the patient and the medical staff must be optimal to deliver the best service. The patient may need help to use the restroom, eat, call someone, or switch on an electrical appliance. If the medical staff receives specific requests regarding a patient's needs, they may act promptly, but most healthcare facilities do not have such an efficient communication system. Primarily the facilities rely on bells and voice commands for communication. These modalities are ineffective when the patient needs to convey a specific message or is unable to speak. Thus, hand-gesture-based interaction seems to be a viable option for communicating the needs. We intend to use hand gestures that can be gestured by simply changing finger arrangements and do not involve much hand movement. These simple hand gestures do not induce much strain on the hand, as the patients can gesture while resting their hands on the bed.

Various techniques can capture patients' hand gestures, such as electromyographs (EMGs), data gloves, and cameras. But EMG and data gloves are cumbersome to use. Therefore, we prefer a vision-based environment where a camera-operated virtual instrument or interface provides the medium for interaction. Moreover, a camera is more readily available and cheaper than other devices for capturing hand gestures. However, barriers to accurate hand gesture recognition include background clutter, human skin regions in the vicinity of the gesturing hand, variations in illumination, and other sources of noise. In this work, we address these barriers and achieve accurate hand gesture recognition using a novel convolution neural network (CNN)–transformer model.

The benefits of CNN lie in its capturing of local information in feature maps, making useful assumptions about the targeted task (i.e., spatial inductive bias), and sharing weights. Transformers, in contrast, excel in fusing global contextual information, generalizing, and including an attention mechanism. Therefore, we propose a two-stage approach to recognizing hand gestures that combines the strengths of both architectures. Its block diagram is shown in Fig. 1. The first stage, that is, saliency detection, accepts the color input image of the hand gesture and generates a saliency map that segments the foreground hand region from the background clutter. The saliency map is a binary image with the hand represented by white

Manuscript received 21 February 2023; revised 3 May 2023; accepted 21 May 2023. Date of publication 5 June 2023; date of current version 23 June 2023. The Associate Editor coordinating the review process was Dr. Zhenghua Chen. *(Corresponding author: H. Pallab Jyoti Dutta.)*

H. Pallab Jyoti Dutta is with the Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India (e-mail: h18@iitg.ac.in).

M. K. Bhuyan is with the Department of Electronics and Electrical Engineering and the Mehta Family School of Data Science and Artificial Intelligence, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India (e-mail: mkb@iitg.ac.in).

Debanga Raj Neog is with the Mehta Family School of Data Science and Artificial Intelligence, Indian Institute of Technology Guwahati, Guwahati, Assam 781039, India (e-mail: dneog@iitg.ac.in).

Karl Fredric MacDorman is with the Luddy School of Informatics, Computing, and Engineering, Indiana University, Indianapolis, IN 46202 USA (e-mail: kmacdorm@indiana.edu).

Rabul Hussain Laskar is with the Department of Electronics and Communication Engineering, National Institute of Technology Silchar, Silchar, Assam 788010, India (e-mail: rhlaskar@ece.nits.ac.in).

Digital Object Identifier 10.1109/TIM.2023.3282655

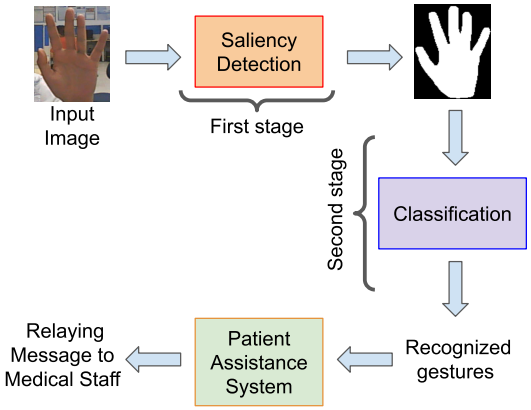


Fig. 1. Block diagram of the proposed method.

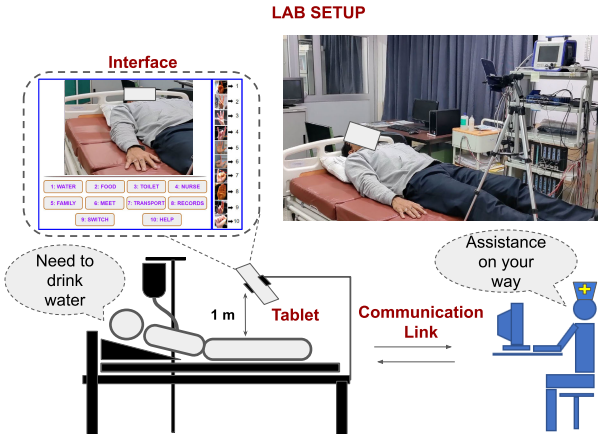


Fig. 2. Schematic of the PAS setup.

pixels and the background by black. The saliency detection stage contains a lightweight encoder–decoder network whose output is fed to the second stage for classification. The second stage labels the recognized class of the hand gesture. This recognized hand gesture relays the patient's message to the medical staff via a patient assistance system (PAS). The PAS is an interface that contains different messages associated with different hand gestures used by a patient to ask for assistance. When a patient gestures, the PAS recognizes the hand gesture and associates the message assigned to it. Then, the PAS transmits the message to the medical staff's computer so that they can view the request and pass it to a staff member who can attend to it.

The PAS's setup consists of a tablet with an in-built camera to capture image frames containing the hand region and display the interface to the patient, a stand to hold the tablet, a communication link (maybe wired or wireless), and a computer to receive the message on the medical staff end. Fig. 2 shows a typical setup. The tablet is placed at a suitable distance so that the interface is visible to the patient, and the patient does not hit it while getting out of bed. The communication link is established over the internet, which may be a wired connection or WiFi. This system has two aspects: one is accurate hand gesture recognition, and the other is optimal transmission management between the end users. This work mainly focuses on hand gesture recognition to develop a robust and efficient system that decodes the messages sent via hand gestures. Now to train the recognition part of the system, we use publicly available hand gesture recognition image datasets captured in complex environments. We used datasets that involve minimal hand movement and can be gestured by simply changing the finger positioning. Using publicly available datasets help us avoid restricting the system to a particular hospital environment but enables its training and testing to generalize to other environments. The result is that the system performs well and overcomes the shortcomings of other vision-based systems. We used HGR [4], [5], HIU [6], and Egohands [7] for the saliency detection stage, and Ouhands [8] and NUS [9] for the classification stage. These datasets contain color images captured in environments that challenge a camera-based recognition system.

The contributions of this work are as follows.

1) We propose a novel saliency detection technique, which combines the benefits of a CNN and a transformer, to obtain a saliency map that precisely separates the hand from the background.
2) We also propose a novel classification network that uses an adaptive kernel channel attention layer (AKCAL) to increase hand gesture recognition accuracy.
3) The transformer's self-attention is replaced with an efficient attention mechanism to reduce computational complexity and memory requirements.
4) A compound saliency loss function is proposed to maintain the hand's geometrical shape and smooth boundary, especially around the fingers. It also corrects for class imbalance.
5) We propose a PAS that translates the patient's hand gestures into messages conveyed to medical staff.
6) The proposed architecture performs well on four benchmark hand segmentation datasets and two hand gesture datasets with better evaluation results than state-of-the-art methods.

This article is organized as follows. Section II reviews previous work in this area. Section III describes the methodology, detailing the saliency detection process, classification stage, and the loss function. Section IV introduces the PAS, and Section V evaluates the proposed method. Finally, Section VI concludes this article.

II. Related Work

Over the years, researchers have been working toward accurate hand gesture recognition due to its pervasive applications in computer vision [1], [3]. Saliency detection is an important step in hand gesture recognition, which frees the image of its background and emphasizes only the foreground. Much work is based on segmenting the hand from the background by machine learning using hand-crafted skin color or texture information and skin modeling algorithms [10], [11], [12]. However, hand-crafted features can introduce bias, and the accuracy of hand mask generation is less than ideal. Recently, improved accuracy from deep learning approaches has gained researchers' attention.

A saliency map can be determined using an encoder–decoder architecture, such as U-Net [13] and Attention U-Net [14]. But generating a precise saliency map to delineate the hand region in the presence of background clutter, skin regions, and varying lighting conditions requires further effort. This area has seen advances such as TransUnet [15]. This transformer-based segmentation model for medical image segmentation combines the benefits of U-Net and transformers. Here, a CNN–transformer-based encoder passes the encoded feature map to a CNN decoder, which performs gradual upsampling. As with U-Net, skip connections from the encoder to the decoder enable precise segmentation results. However, this approach requires more transformer units, resulting in more training parameters and greater computational complexity. In [16], the segmentation architecture was fine-tuned for hand segmentation on first-person videos. Conditional random fields further refine the segmented maps to achieve precise results for various datasets. Cai et al. [17] modeled prediction uncertainty across various domains and the hand shape information. They also considered the first-person view and performed hand segmentation using a self-supervised Bayesian-CNN. In [18], segmentation masks are generated for depth videos by combining two soft proposals a proposal detecting the hand using a CNN on the current frame and another proposal tracking the hand using a Kalman filter from the previous frame.

Once the saliency map has been determined, this information is passed to the recognition network. Restricting the input to the region of interest improves recognition rates. Dadashzadeh et al. [1] proposed a network that segments the hand using residual convolution blocks (RCBs) and atrous convolutions. The original input and the segmented mask form a two-way network whose final layer predicts the output of the hand gesture. Dutta et al. [19] adopted a similar approach, that is, labeling 34 classes of hand gestures after segmenting the hand region using U-Net. However, the images contained uniform backgrounds. Bao et al. [20] recognized hand gestures without localizing the hand and achieved good results on images with a simple background. Their performance on images with complex backgrounds was average, and most importantly, they considered only seven classes. Chevtchenko et al. [21] proposed a three-way feature extraction scheme to obtain hand gesture recognition. They concatenated features obtained from a CNN with the original input image, a CNN with Gabor filter output as input, and hand-crafted Zernike moments, Hu moments, and contour features of the hand. This makes the entire network a bit messy, which can be avoided as the goal of deep neural networks is to avoid hand-crafted features. Le et al. [22] proposed a multiscale region-based fully convolution network that performs hand detection and classification. They used different filter sizes to capture global and local features and achieve multiscale feature extraction. This also highlights the usefulness of precise localization and classification for accurate hand gesture recognition in complex surroundings.

Despite much work on hand gesture recognition, researchers have yet to achieve a robust technique to obtain accurate results in challenging environments, as highlighted earlier. This work targets these issues by proposing a two-step approach shown effective through evaluation with benchmark datasets.

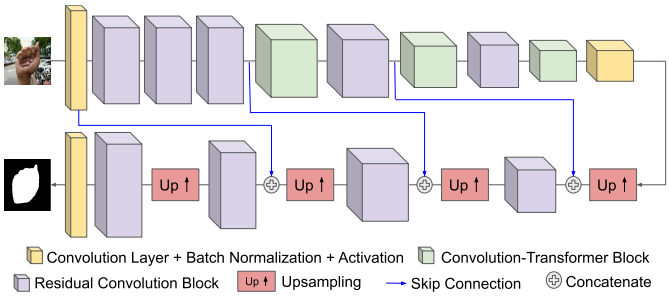


Fig. 3. Encoder–decoder architecture for saliency detection.

III. Methodology

This section describes the proposed methodology for saliency map estimation and saliency-map-based hand gesture recognition. It also explains the procedure for operating a PAS using recognized hand gestures. The key components of the proposed method are described in Sections III-A–III-B.

A. Saliency Detection

Saliency detection is a method to determine the region of interest in a color image. In our case, this preprocessing phase segments the hand from the background to remove irrelevant content, such as background clutter, and negates the effect of illumination variations through pixel-by-pixel binary labeling of the foreground and background. The proposed saliency detection uses an encoder that accepts a color image as input and encodes it in a compressed latent space. The encoder is a combination of convolution and transformer blocks to embed the local and finer semantics of the object and the long-range (global) relationships in an image. The convolution block is arranged at the beginning to encode the structural and shape features, which results in a compact representation of the input. This compact representation is fed to the transformer, which further encodes it into object-specific high-level features. These high-level features do not convey any meaning visually but possess semantically significant attributes for classification tasks. Moreover, transformers contain multihead self-attention mechanisms (SAMs). If these mechanisms are placed toward the end stage, it improves prediction performance [23]. Thus, they are added to the later part of the encoder. The latent feature maps generated by the encoder are fed to the decoder, gradually upsampling the feature maps to match the dimension of the input image. After each upsampling, the decoder's resultant feature maps are concatenated with the encoder's corresponding dimension feature maps to obtain precise segmentation results. The decoder has no transformers because it only decodes whatever has been encoded without the need to capture the high-level abstract representation. The block diagram of the encoder–decoder architecture is shown in Fig. 3.

An input image, $I \in \mathbb{R}^{H\times W\times C}$, is given to a block of convolution, batch normalization, and ReLU layers. The resultant

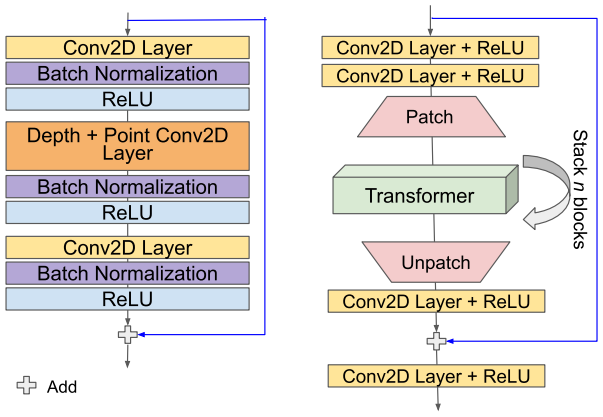


Fig. 4. Left: the RCB. Right: the CTB.

feature map, $F \in \mathbb{R}^{H' \times W' \times C'}$, is passed through three blocks of RCBs, shown in Fig. 4 (left), and the output is given as

$$F' = (\mathcal{R} \circ \text{BN} \circ \text{conv} \circ \mathcal{R} \circ \text{BN} \circ \text{DP}_{\text{conv}} \circ \mathcal{R} \circ \text{BN} \circ \text{conv})(F) + F \quad (1)$$

where BN is the batch normalization layer, conv is the convolution layer, $\mathcal{R}$ is the ReLU activation, and DP_{conv} is the depth + pointwise convolution layer. The representation $(f \circ g)(x)$ in the equation means $f(g(x))$. The depth + pointwise convolution layer adds fewer parameters to a model's training parameters than a convolution layer. Thus, it is an efficient means of learning a feature map's representation. The addition of F in (1) represents skip connections to counteract vanishing gradients. The downsampled F' is fed to a convolution–transformer block (CTB), shown in Fig. 4 (right), which embeds the spatial inductive bias and distal pixel dependencies. CTB's output is given as

$$f = (\mathcal{R} \circ \text{conv})\{(\mathcal{R} \circ \text{conv} \circ \mathcal{U} \circ \text{Transformer}^{(n)} \circ \mathcal{P} \circ \mathcal{R} \circ \text{conv} \circ \mathcal{R} \circ \text{conv})(F') + F'\}. \quad (2)$$

Here, $\mathcal{P}$ represents dividing the feature map F'' from the second convolution layer of CTB into patches, i.e., $\mathcal{P} : F'' \rightarrow F'''$, $F''' \in \mathbb{R}^{(hw) \times \eta \times C''}$, which imitate the sequential arrangement to be given to the transformer unit. The height and width of each patch are denoted by h and w, respectively. C'' is the channel number of F'' and $\eta = (H''W'')/(hw)$ with H'' and W'' being the height and width of F'', respectively. The transformer unit, represented by $\text{Transformer}^{(n)}$, is stacked n times before passing the feature map to the "unpatching" operation, $\mathcal{U}$, which maps the transformer output F_{trans} to a convolution-like feature map, i.e., $\mathcal{U} : F_{\text{trans}} \rightarrow F'''', F'''' \in \mathbb{R}^{H'' \times W'' \times C''}$.

The encoder contains a CTB after the third, fourth, and fifth RCBs, followed by a block of convolution, batch normalization, and ReLU layers, as shown in Fig. 3. The encoder's last convolution block maps its output to a high-dimensional latent space, from which the feature map f is fed to the decoder. The decoder is a sequential arrangement of convolution blocks represented as

$$f_{\text{saliency}} = (\text{sigmoid} \circ \text{BN} \circ \text{conv} \circ \text{RCB} \circ \text{Up} \uparrow \circ \text{RCB} \circ \oplus^3 \text{Up} \uparrow \circ \text{RCB} \circ \oplus^2 \text{Up} \uparrow \circ \text{RCB} \circ \oplus^1 \text{Up} \uparrow)(f). \quad (3)$$

TABLE I

DESCRIPTION OF THE SYMBOLS USED FOR THE SALIENCY DETECTION AND CLASSIFICATION ARCHITECTURE

Category	Symbol	Description
Layers	conv	Convolution layer
	BN	Batch normalization layer
	DP_{conv}	Depth + point convolution layer
	$\mathcal{R}$	ReLU activation
	$\mathcal{P}$	Converts the feature map into patches to be provided to transformer as input
	$\mathcal{U}$	Mapping of transformer output to convolution-like map
	$\text{Transformer}^{(n)}$	Stack n transformer layers sequentially
	Up↑	Upsampling layer
	sigmoid	Sigmoid activation
	RCB	Residual convolution block
	LN	Layer normalization layer
	$D_{0.1}$	Dropout layer with dropout rate = 0.1
	$\mathcal{D}$	Dense layer
	$\text{attention}_{\text{new}}$	Attention map from the attention module in the transformer unit
	$\mathcal{M}$	Max pooling layer
	AKCAL	Adaptive kernel channel attention layer
	CTB	Convolution transformer block
operation	$\circ$	$(f \circ g)(x) = f(g(x))$ which means applying the function g on x and then the function f on $g(x)$
	$\oplus$	Skip-connections from the encoder to the decoder
	$\otimes$	Element-wise multiplication

Up ↑ is the upsampling of the incoming feature map by a factor of 2. $\oplus^1$, $\oplus^2$, and $\oplus^3$ are the skip connections from the encoder's fourth RCB, third RCB, and first ReLU activation layer, respectively. The sigmoid activation layer outputs the salient feature map f_{saliency} emphasizing the hand region and removing the background. Many symbols are used to describe the architecture's layers and operation, tabulated in Table I.

Transformer Unit: The transformer unit proposed here does not use the SAM generally used by transformers. The proposed transformer unit linearizes the quadratic self-attention by ending the quadratic dependency on the spatial dimension of the patch. Thus, the computational complexity and memory requirements are reduced significantly. The block diagram of the transformer unit is shown in Fig. 5.

Transformers encode an image's global characteristics using a multihead attention mechanism that combines several SAMs [24]. In an SAM, a feature map $\mathcal{F} \in \mathbb{R}^{m \times n \times c}$ is constructed into $\mathcal{F}^1 \in \mathbb{R}^{mn \times c}$ and for ath location in $\mathcal{F}^1$, $\exists\, \mathcal{G}_a \in \mathbb{R}^c$. From $\mathcal{G}_a$, three vectors $\{\mathcal{G}_a^q, \mathcal{G}_a^k, \mathcal{G}_a^v\} \in \mathbb{R}^{d_k}$ are generated: query, key, and value. The dot product $(\mathcal{G}_a^q)^T \mathcal{G}_b^k$ measures the relevance of the present location for attention, i.e., the significance of the bth location with respect to the ath location. This dot product is stabilized by $(d_k)^{1/2}$ and normalized by the softmax function to guarantee positive values. Finally, the product is multiplied by $\mathcal{G}_a^v$ to acquire the weighted value vector at a. Similarly, for all the mn locations, the weighted value vectors are calculated and summed to obtain the SAM's output attention vector at the ath location. In terms of matrices, the query, key, and value matrices are represented as $\{\mathcal{Q}, \mathcal{K}, \mathcal{V}\} \in \mathbb{R}^{mn \times d_k}$. Therefore, the output attention map is

$$\text{Output attention map} = \text{softmax}\left(\frac{\mathcal{Q}\mathcal{K}^T}{\sqrt{d_k}}\right)\mathcal{V}. \quad (4)$$

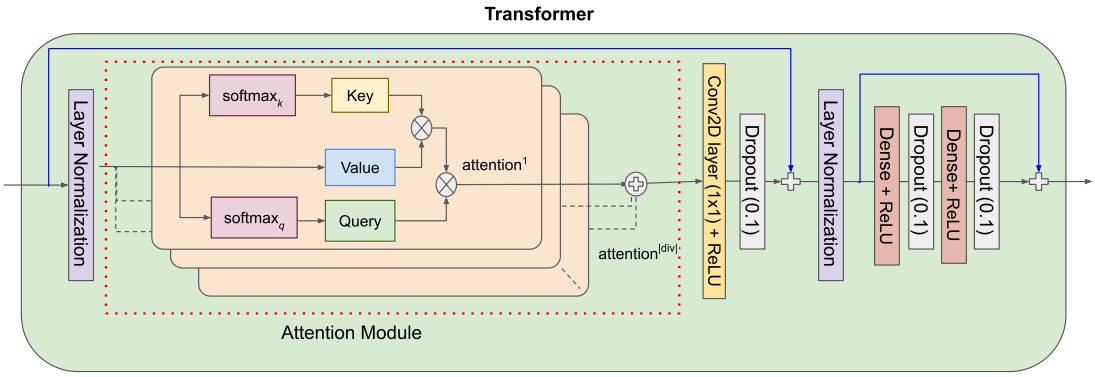


Fig. 5. Transformer unit.

However, the SAM has $\mathcal{O}(d_k(mn)^2)$ computational complexity. Moreover, for systems with limited memory, its $\mathcal{O}((mn)^2)$ memory usage can result in an out-of-memory error. Thus, inspired by Zhuoran et al. [25], we propose a transformer unit with linear, instead of quadratic complexity. Here, the queries, keys, and values are grouped, such that the respective matrices are $\{\mathcal{Q}^i, \mathcal{K}^i, \mathcal{V}^i\} \in \mathbb{R}^{mn\times(d_k)/(|\text{div}|)}$. i representing the ith group, $i \in [1, \ldots, |\text{div}|]$ and $|\text{div}|$ is the number of divided groups. Now

$$(\mathcal{QK})^T\mathcal{V} = \mathcal{Q}\left(\mathcal{K}^T\mathcal{V}\right) \quad \text{(Matrix associativity)}. \tag{5}$$

This associativity with softmax function can be achieved using two softmax operations: 1) query softmax and 2) key softmax [25]. The resulting attention map for the ith group is

$$\text{attention}^i = \text{softmax}_q\left(\mathcal{Q}^i\right)\left(\text{softmax}_k\left(\mathcal{K}^i\right)^T\mathcal{V}^i\right). \tag{6}$$

Similarly, for all i we have attention^1, $\text{attention}^2, \ldots,$ $\text{attention}^{\#\text{div}}$, which are concatenated and feed to a convolution layer to obtain a new attention map given by

$$\text{attention}_{\text{new}} = \text{conv}\left(\left[\text{attention}^1, \text{attention}^2, \ldots, \text{attention}^{|\text{div}|}\right]\right) \in \mathbb{R}^{mn\times c}. \tag{7}$$

This attention map treats each channel of the key as a feature map. The channel is multiplied by value to compute global features that weight every location of the incoming feature map. The global features are accumulated by multiplying query with the product of key and value. The final product ensures attention to class-specific features and their location.

Due to this modification, the memory complexity of the transformer unit is reduced from $\mathcal{O}((mn)^2)$ to $\mathcal{O}(d_kmn + d_k^2)$ and computational complexity from $\mathcal{O}(d_k(mn)^2)$ to $\mathcal{O}((d_k)^2mn)$. Thus, the quadratic increase in complexity with the feature map's spatial dimensions is avoided. The complexity now depends on d_k, which is set by the developer.

The sequential features from the patch block are fed to the layer normalization layer of the transformer unit. Since layer normalization deals with sequential data better than batch normalization and is independent of batch size, it is adopted for the transformer's normalization [26]. The transformer's output feature map, f_{trans} is given by

$$\begin{aligned} f_{\text{int}} &= \text{LN}[(D_{0.1} \circ \mathcal{R} \circ \text{attention}_{\text{new}} \circ \text{LN})(\mathcal{P}) + \mathcal{P}] \\ f_{\text{trans}} &= (D_{0.1} \circ \mathcal{R} \circ \mathcal{D} \circ D_{0.1} \circ \mathcal{R} \circ \mathcal{D})(f_{\text{int}}) + f_{\text{int}}. \end{aligned} \tag{8}$$

Here, LN is the layer normalization layer, $D_{0.1}$ is the dropout layer with the dropout rate set to 0.1, and $\mathcal{D}$ is the dense layer. f_{int} is the intermediate feature map that is given to the feedforward network of the transformer. Two skip connections facilitate transfer of finer details to the later layers.

B. Classification

The saliency map, after the removal of background clutter, is fed to the proposed classification network. The highlights of the classification network are the novel convolution–transformer feature engineering model and AKCAL. AKCAL uses adaptive kernels at different scales to ensure automatic feature extraction, ensuring the network focuses on only the hand region, not on the adjoining background region. This enhances gesture recognition accuracy.

The saliency map passes through a convolution-batch normalization-ReLU block and a max pooling layer. This feature map is fed to a sequential arrangement of three blocks: 1) the depth + point convolution layer; 2) the convolution-batch normalization-ReLU block; and 3) AKCAL. Next, the input is added as a skip connection to the output. A max pooling layer follows, which halves the spatial dimension. This sequence of passing the feature map through a sequential arrangement of feature extracting blocks, and skip connection followed by a max pooling layer is carried out for three more levels before passing to a multilayer perceptron (MLP) for classification. Toward the end of the network, CTB replaces the convolution–batch normalization–ReLU block because of the transformer's ability to encode high-level class-specific features. The following set of equations describes the feature engineering approach:

$$\Phi_1 = (\mathcal{M} \circ \mathcal{R} \circ \text{BN} \circ \text{conv})\left(f_{\text{saliency}}\right) \tag{9}$$

$$\Phi_2 = \mathcal{M}[(\text{AKCAL} \circ \mathcal{R} \circ \text{BN} \circ \text{conv} \circ \text{DP}_{\text{conv}})(\Phi_1) + \Phi_1] \tag{10}$$

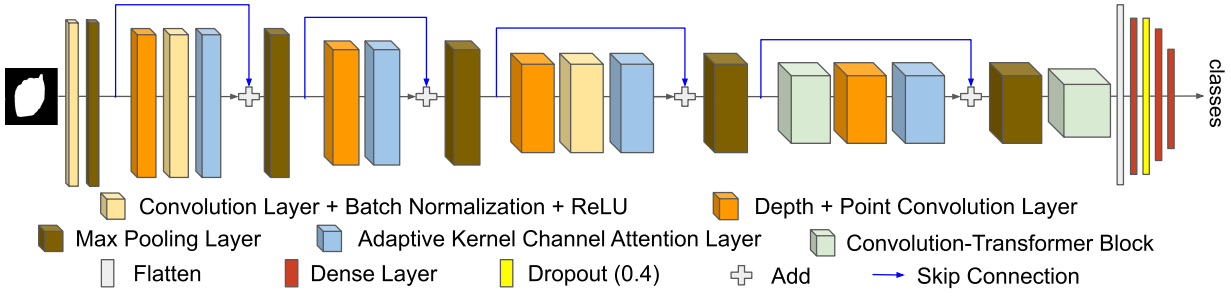
$$\Phi_3 = \mathcal{M}[(\text{AKCAL} \circ \text{DP}_{\text{conv}})(\Phi_2) + \Phi_2] \tag{11}$$

$$\Phi_4 = \mathcal{M}[(\text{AKCAL} \circ \text{BN} \circ \text{conv} \circ \text{DP}_{\text{conv}})(\Phi_3) + \Phi_3] \tag{12}$$

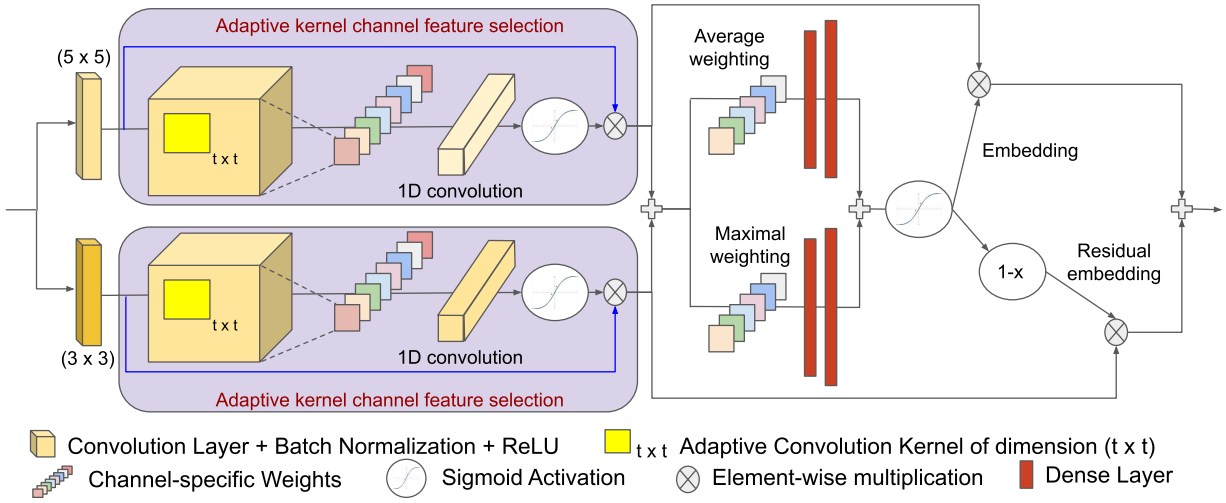
$$\Phi_5 = (\text{CTB} \circ \mathcal{M})[(\text{AKCAL} \circ \text{DP}_{\text{conv}} \circ \text{CTB})(\Phi_4) + \Phi_4] \tag{13}$$

where $\mathcal{M}$ denotes the max pooling layer. The feature map Φ_5 is flattened and passed to the classifier given by

$$\text{classes} = (\mathcal{D} \circ \mathcal{D} \circ D_{0.4} \circ \mathcal{D} \circ \text{flatten})(\Phi_5). \tag{14}$$



(a)



(b)

Fig. 6. Classification module. (a) Classification network. (b) AKCAL.

The number of neurons in the MLP's hidden layers is progressively reduced to lower the parameter count. The block diagram for the classification network is shown in Fig. 6(a) and the AKCAL is shown in Fig. 6(b).

AKCAL: The incoming feature map, $\mathcal{X} \in \mathbb{R}^{x \times y \times z}$, is divided into two feature maps, $\mathcal{X}_1$ and $\mathcal{X}_2 \in \mathbb{R}^{x \times y \times z}$, encoded via 3×3 and 5×5 kernels, respectively. The goal of this division is to extract different scale information using 3×3 and 5×5 kernels. Increasing the kernel size increases its receptive area to capture more information from a region but at the cost of more training parameters. Therefore, only two divisions are considered with different kernel sizes.

Next, feature maps from both the paths are passed through an adaptive kernel with a channel feature selection module, which starts with a convolution-batch normalization-ReLU block. The kernel $t \times t$ for the convolution layer is adaptive so that the extent of feature interactions is not determined manually. Adjusting the parameter count allows for optimal encoding of features. Since the class-specific features are encoded across channels, we consider a function that defines a relationship between channel Z and kernel size t. Moreover, the number of channels is a power of 2; therefore, the relationship is defined as

$$Z = 2^{(\rho t)} \Rightarrow t = \frac{\log_2 z}{\rho}, \quad \rho \text{ is constant.} \tag{15}$$

Subsequently, the module encodes the global features of each channel via an average weighting scheme given by

$$Z_1^l = \frac{1}{xy} \sum_{u=1}^{x} \sum_{v=1}^{y} \mathcal{X}_{t \times t}^l(u, v). \tag{16}$$

$\mathcal{X}_{t \times t}^l(u, v)$ represents the lth feature map after the adaptive kernel convolution layer. A 1-D convolution layer further encodes this with an adaptive kernel t and sigmoid activation. The resulting attention map is multiplied with the input to this module to emphasize the channel features and outputs the features $\mathcal{X}_1'$ and $\mathcal{X}_2'$ for the two paths, respectively. $\mathcal{X}_1'$ and $\mathcal{X}_2'$ are added and passed through two paths that weight them (using (16) and (17), respectively) to generate channel-specific weights

$$Z_2^l = \max_{u \in \{1,\ldots,x\}, v \in \{1,\ldots,y\}} \mathcal{X}_{t \times t}^l(u, v). \tag{17}$$

The inclusion of both average and maximal weighting increases classification accuracy because average weighing captures the feature map's soft global characteristics and maximal weighting captures their most significant characteristics. The weighted features are then passed through two dense layers and their outputs are added together to form an intermediate feature map $\mathcal{X}_{\text{int}}$. $\mathcal{X}_{\text{int}}$ is passed through a sigmoid layer to generate the embedding and residual embedding. These embeddings select classification-relevant channel features. The

output of the AKCAL is

$$f_{\text{cal}} = \left(\text{embedding} \otimes \mathcal{X}_1'\right) + \left(\text{residual embedding} \otimes \mathcal{X}_2'\right) \quad (18)$$

where embedding is the weighted and encoded output of the adaptive kernel channel feature selection module. It is the output of AKCAL's last sigmoid activation. Residual embedding represents the residual weighted and encoded output obtained by subtracting embedding from 1. $\otimes$ denotes elementwise multiplication.

C. Loss Function

The proposed architecture uses two loss functions to optimize the saliency detection and classification models, as explained below.

1) Loss Function for Saliency Detection: The proposed saliency detection loss function is a compound function that segments the hand region, assures its edge continuity and smoothness, and compensates for foreground–background class imbalance.

If binary cross entropy (BCE) is used to segment the hand, categorization is biased toward its background because the background contains far more image pixels. To compensate for this bias, we use binary focal loss (BFL) [27]. This reduces the weight of easily classifiable background pixels in optimization and increases the weight of foreground pixels. The BFL is given by

$$\text{BFL}\left(y^p, y^t\right) = \begin{cases} -\lambda_1\left(1-y^p\right)^{\lambda_2}\log\left(y^p\right), & y^t = 1 \\ -(1-\lambda_1)\left(y^p\right)^{\lambda_2}\log\left(1-y^p\right), & \text{otherwise.} \end{cases} \quad (19)$$

y^p and y^t stand for the predicted and target probabilities, respectively. λ_1 compensates for the class imbalance, and λ_2 reduces the contribution of background pixels and increases that of foreground pixels.

Moreover, delineating the hand region is essential for accurate segmentation, especially for finger regions. The finger regions are narrow and easily affected by background clutter and skin areas, leading to rough edges along the boundary. This must be smoothed for precise delineation by energy reduction along the hand edges. Thus, we propose a loss function that smooths the edges, given by

$$\mathcal{L}_{\text{smoothing}} = \sum_{(i,j)\in\mathcal{I}} \sqrt{\left|\left(\nabla y^p_{x_{\text{ij}}}\right)^2 + \left(\nabla y^p_{y_{\text{ij}}}\right)^2\right|} \quad (20)$$

where $\nabla y^p_{x_{\text{ij}}}$ and $\nabla y^p_{y_{\text{ij}}}$ represent the gradient along the horizontal and vertical directions, respectively. The index set $\mathcal{I}$ contains the coordinates of the elements of y^p.

The shape information of the hand is also crucial. Hand skeleton information helps maintain its shape's integrity by preventing hand parts from being segmented owing to occlusion, slender foreground regions, and background clutter. To ensure hand structure connectedness, we use centerline dice [28] together with the Dice coefficient, which measures the resemblance between the saliency mask and true mask. Here, the skeletons of the saliency mask ($\mathcal{S}_\mathcal{P}$) and true mask ($\mathcal{S}_\mathcal{T}$) are calculated, and the intersection of the two masks is determined. This intersection is iteratively improved and helps assure structural connectedness of different parts of the hand. The loss is mathematically represented as

$$\mathcal{L}_{\text{skdice}} = \alpha_1\left(1 - \frac{2|y^p \otimes y^t|}{|y^p| + |y^t| + \epsilon}\right) + \alpha_2\left(1 - \frac{2\left(\frac{|\mathcal{S}_\mathcal{P}\otimes y^t|}{|\mathcal{S}_\mathcal{P}|+\epsilon} \times \frac{|\mathcal{S}_\mathcal{T}\otimes y^p|}{|\mathcal{S}_\mathcal{T}|+\epsilon}\right)}{\frac{|\mathcal{S}_\mathcal{P}\otimes y^t|}{|\mathcal{S}_\mathcal{P}|+\epsilon}}\right) \quad (21)$$

where $\alpha_1 = \alpha_2 = 0.5$, $\otimes$ indicates elementwise multiplication, and $|.|$ represents the set's cardinality.

The proposed combined loss functions is given by

$$\mathcal{L}_{\text{saliency}} = \text{BFL} + \mathcal{L}_{\text{smoothing}} + \mathcal{L}_{\text{skdice}}. \quad (22)$$

2) Loss Function for Classification: Categorical cross-entropy is used for classification loss, given by

$$\mathcal{L}_{\text{class}} = \sum_{j=1}^{|\text{classes}|} \sum_{i=1}^{|\text{samples}|} y^t_{\text{ij}} \log y^p_{\text{ij}}. \quad (23)$$

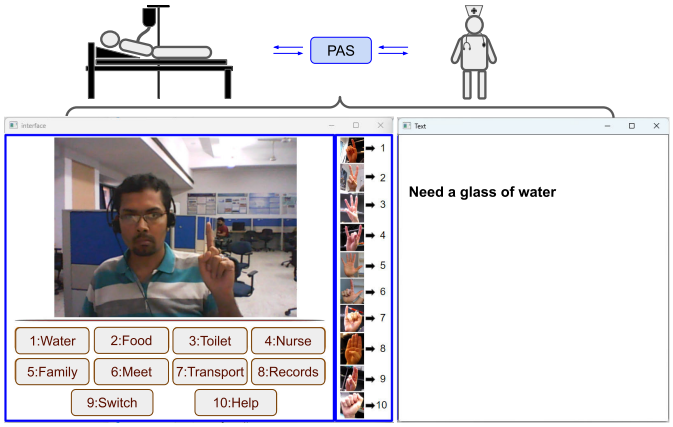


Fig. 7. PAS.

IV. Patient Assistance System

The PAS is a novel application of the proposed method, which lets patients convey specific messages to the medical staff when they need assistance. The PAS's interface is shown in Fig. 7, and the setup is shown in Fig. 2. This system is convenient for patients who are bedridden or otherwise lack mobility. It is operated by hand gestures that involve simple finger arrangements. The ten available gestures are visible on the interface to help patients with gesturing. They correspond to the ten classes in the dataset on which the classification method was trained and the ten messages, shown in Fig. 8.

The patients gesture the specific hand gesture according to their needs, and the PAS recognizes it. The PAS captures the patient's hand gesture through the in-built camera of a tablet mounted on a stand above the bed the patient is lying on. The tablet displays the interface where the patient can get the live feed of gesturing. Once the patient gestures, the PAS's interface runs the proposed saliency detection algorithm to transform the color image into a binary image containing the hand and background and passes it to the proposed classification stage to obtain the class label. All this is done

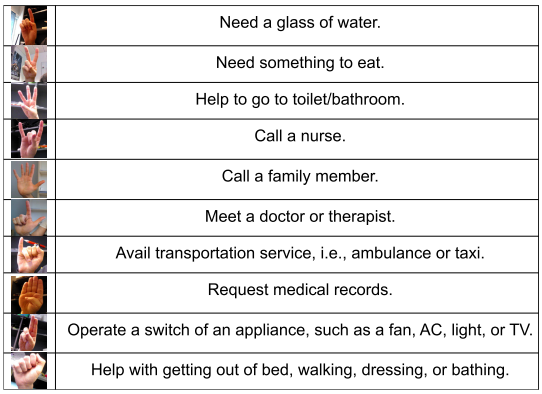


Fig. 8. Messages and their corresponding gestures.

TABLE II
TRAINING DETAILS

Environment	Python 3.6
GPU	Nvidia Tesla P100
Optimizer	Adam Optimizer
Learning rate	0.0001
α and β (for BFL)	0.25 and 2 (from [27])
Batch size	8 for saliency detection and 16 for classification
Epochs	20 for saliency detection and 30 for classification
Division of data	70% training, 15% validation, 15% testing
$d_k = d_v$	[16, 64] for encoder [128, 32] for decoder

in the processing unit of the tablet/computer. Finally, the message assigned to the recognized gesture is transmitted via the wired or wireless communication link to the medical staff's computer. Upon receiving the message, it is forwarded to the staff member who can address it. Simultaneously, the medical staff acknowledges it by text, which the system converts into speech and plays in the patient's room. This assures patients that their requests are being attended to. Moreover, there is a bell attached to the system for the message "SOS" when the patient needs immediate attention. This is a more effective way to communicate than pressing a call button because the patient can convey different messages.

V. EVALUATION AND RESULTS

This section reports the evaluation of the proposed architecture based on the quantitative and qualitative results. The datasets are also described. Training information is given in Table II. The F1-score and mean intersection over union (mIoU) were adopted as the performance measure for saliency detection and accuracy for classification. They are defined as follows:

$$\text{F1-score} = \frac{2}{|\text{cls}|} \frac{\sum_{i=1}^{|\text{cls}|} \frac{\mathcal{A}_{\text{ii}}}{\sum_{j=1}^{|\text{cls}|} \mathcal{A}_{\text{ji}}} \times \sum_{i=1}^{|\text{cls}|} \frac{\mathcal{A}_{\text{ii}}}{\sum_{j=1}^{|\text{cls}|} \mathcal{A}_{\text{ij}}}}{\sum_{i=1}^{|\text{cls}|} \frac{\mathcal{A}_{\text{ii}}}{\sum_{j=1}^{|\text{cls}|} \mathcal{A}_{\text{ji}}} + \sum_{i=1}^{|\text{cls}|} \frac{\mathcal{A}_{\text{ii}}}{\sum_{j=1}^{|\text{cls}|} \mathcal{A}_{\text{ij}}}} \tag{24}$$

$$\text{mIoU} = \frac{1}{|\text{cls}|} \sum_{i=1}^{|\text{cls}|} \frac{\mathcal{A}_{\text{ii}}}{\sum_{j=1}^{|\text{cls}|} \mathcal{A}_{\text{ij}} + \sum_{j=1}^{|\text{cls}|} \mathcal{A}_{\text{ji}} - \mathcal{A}_{\text{ii}}} \tag{25}$$

$$\text{Accuracy} = \frac{\sum_i \mathcal{A}_{\text{ii}}}{\sum_{i,j} \mathcal{A}_{\text{ij}}} \tag{26}$$

where $\mathcal{A}$ denotes the confusion matrix, and $|\text{cls}|$ denotes the number of classes. Average precision is given by $(1)/(|\text{cls}|) \sum_i (\mathcal{A}_{\text{ii}})/(\sum_j \mathcal{A}_{\text{ji}})$ and average recall by $(1)/(|\text{cls}|) \sum_i (\mathcal{A}_{\text{ii}})/(\sum_j \mathcal{A}_{\text{ij}})$.

A. Datasets

Five datasets were used in the proposed work: 1) Ouhands [8]; 2) NUS [9]; 3) HGR [4], [5]; 4) HIU [6]; and 5) Egohands [7]. However, Ouhands and NUS contain labels for classification and HGR, HIU, and Egohands do not. Therefore, we evaluate saliency detection using the Ouhands, HGR, HIU, and Egohands datasets and classification using the Ouhands and NUS datasets. The five datasets are described below.

The Ouhands static hand gesture dataset comprises color images, segmentation masks, depth images, and hand bounding box annotations for ten classes. The dataset consists of 3000 color images contributed by 23 subjects.

The HGR dataset combines three datasets: 1) HGR1; 2) HGR2A; and 3) HGR2B. The first dataset contains 899 images and 25 classes, the second dataset contains 85 images and 13 classes, and the third contains 574 images and 32 classes. The datasets include segmentation masks and hand-joint locations for each image.

NUS is a set of datasets with ten classes. NUS I comprises images with uniform backgrounds and is thus excluded. However, NUS II comprises images with all the challenges encountered in hand gesture recognition. It contains 2000 color images and another 750 images that include human skin regions other than the hands.

The HIU dataset contains 33 000 color images, corresponding hand segmentation masks, and hand-joint locations.

Egohands is a segmentation dataset with 48 videos taken from a first-person viewpoint. There are 4800 annotated frames containing multiple hands, which complicates hand segmentation.

Every dataset contains challenges, such as occlusion, background clutter, and varying lighting conditions and pose angles. Image resolution varies by dataset and is scaled to a uniform spatial dimension before feeding the image to the model. For the ablation study, we use the Ouhands dataset unless it is specified otherwise.

B. Experiments and Results

Table III reports the effect of dividing the $\mathcal{Q}$, $\mathcal{K}$, and $\mathcal{V}$ into groups of different sizes on the saliency detection model's performance. Based on observation, the division of the matrices into two groups reports the maximum F1-score. Since this resulted in better saliency maps, two groups were selected even though three groups resulted in a slightly lower inference time.

The contribution of the loss functions in optimizing the saliency detection model is shown in Table IV. The BFL produced a better F1-score than BCE, so it was used to combine the other losses. The table shows that the combination of $\mathcal{L}_{\text{smoothing}}$ and $\mathcal{L}_{\text{skdice}}$ with BFL produced the best result. Thus, this combination is retained.

Moreover, the proposed saliency detection model was compared with a few baseline models, as shown in Table V. The proposed saliency detection method performs better than the

TABLE III

PERFORMANCE OF THE SALIENCY DETECTION MODEL FOR DIFFERENT NUMBERS OF GROUPS OF $\mathcal{Q}$, $\mathcal{K}$, AND $\mathcal{V}$

Number of groups	F1-score (%)	Inference (ms)
1	96.47	17.8
2	**97.33**	15.03
3	96.20	**14.4**
4	96.01	14.9
5	95.86	14.8

TABLE IV

PERFORMANCE OF THE PROPOSED SALIENCY DETECTION MODEL WITH VARIOUS LOSS FUNCTIONS

Losses	F1-score (%)
BCE	96.49
BFL	96.93
BFL + $\mathcal{L}_{\text{smoothing}}$	97.10
BFL + $\mathcal{L}_{\text{skdice}}$	97.25
BFL + $\mathcal{L}_{\text{smoothing}}$ + $\mathcal{L}_{\text{skdice}}$	**97.33**

TABLE V

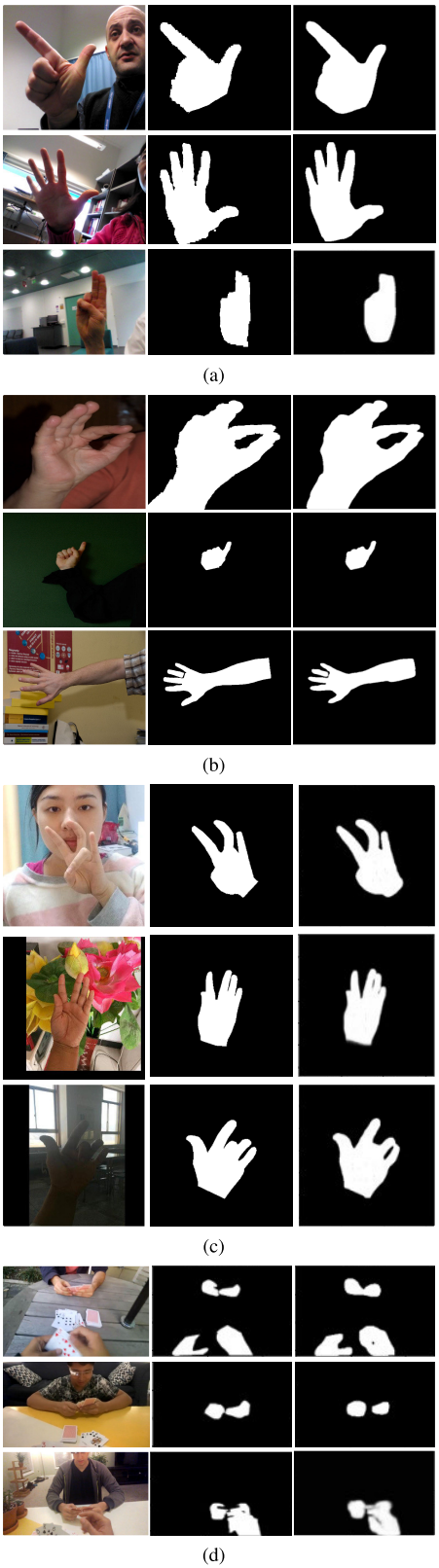
COMPARISON OF DIFFERENT BASELINE MODELS WITH THE PROPOSED SALIENCY MODEL

Methods	F1-score (%)	mIoU (%)	Inference time (ms)	#Parameters
FCN-8s [29]	95.5	80.3	63	134 M
Attention U-Net [14]	96.8	74.8	29	6.44 M
RAU-Net [30]	96.9	83.8	32	11.62 M
DeepLabv3 [31]	97.1	87.3	43	75.30 M
DeepLabv3+ [32]	**97.3**	88.5	56	85.12 M
HGR-Net [1]	96.3	82.62	21	**0.28 M**
CBAM [33]	96.3	82.91	38	7.97 M
DANet [34]	96.4	83.69	35	7.56 M
U-Net [13]	96.7	81.26	25	7.86 M
U-Net + Triplet Attention [35]	85.4	70.18	25	7.88 M
U-Net + CondConv [36]	92.7	75.39	31	9.7M
PSPNet [37]	95.7	80.2	18	0.96 M
ICNet [38]	95.3	81.6	24	6.83 M
HRNet [39]	96.8	88.5	38	28.60 M
SegNet [40]	96.1	80.6	27	2.95 M
Ours	**97.3**	**89.0**	**15**	0.88 M

other techniques. Its F1-score, mIoU, and inference time are the best among the compared models, and its parameter count is also low, second only to HGR-Net.

The qualitative result of the saliency detection method for four datasets is shown in Fig. 9. The results indicate that the saliency maps are robust despite the variations in illumination, occlusion, and the presence of skin regions and background clutter. Moreover, they produce good segmentation results for multiple hands. Thus, this preprocessing step is efficient and effective and can be combined with the classification step without substantially increasing memory complexity.

The recognition rate of the proposed method is high with an accuracy of 93.8% and 98.0% for Ouhands and NUS II, respectively. The use of the saliency map helped achieve phenomenal results because it mitigated the major challenges encountered in hand gesture recognition. As shown in Table VI, using a pretrained VGG16 [41] network on saliency detected images yielded better results than on RGB images. Similarly, an architecture similar to the encoder part of the saliency detection method (termed as encoder) performed better with saliency maps than color images. This indicates the benefits of using saliency maps. Moreover, we tried different combinations, such as attaching different feature-enhancing modules to the encoder. The performance is recorded in



(a) (b) (c) (d)

Fig. 9. Saliency maps for different datasets. The first column shows the color images, the second column shows the ground truth, and the third column shows the detected salient regions. (a) Saliency map for the Ouhands dataset. (b) Saliency map for the HGR dataset. (c) Saliency map for the HIU dataset. (d) Saliency map for the Egohands dataset.

TABLE VI

PERFORMANCE COMPARISON OF BASELINES WITH DIFFERENT COMBINATIONS OF SUPPORTING MODULES

Models	Accuracy (%)
pre-trained VGG16 [41] (RGB images)	72.8
pre-trained VGG16 [41] (saliency maps)	81.6
Encoder (RGB images)	85.6
Encoder (saliency maps)	87.1
Encoder + Triplet Attention [35]	72.4
CondConv Encoder [36]	84.2
Adaptive Kernel Encoder	89.8
Channel Attention Encoder	91.0
Adaptive Kernel Channel Attention Layer Encoder	**93.8**

TABLE VII

PERFORMANCE FOR DIFFERENT VALUES OF RHO TO DETERMINE OPTIMAL ADAPTIVE KERNEL SIZE t

ρ values	Accuracy (%)
ρ = 0.5	84.2
ρ = 1	87.2
ρ = 1.5	89.6
ρ = 2	**93.8**
ρ = 2.5	90.2

TABLE VIII

PERFORMANCE COMPARISON FOR DIFFERENT WEIGHTING SCHEMES IN AKCAL

Weighting schemes	Accuracy (%)
no weighting	86.0
only average weighting	88.7
only max weighting	89.8
both average and max weighting	**93.8**

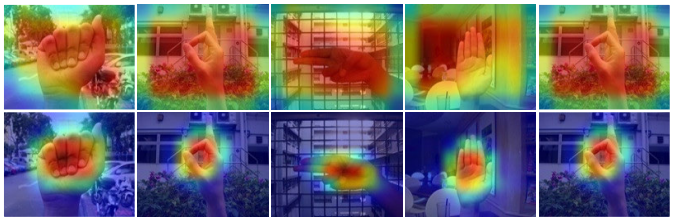


Fig. 10. Effect of AKCAL. Top: without AKCAL. Bottom: with AKCAL.

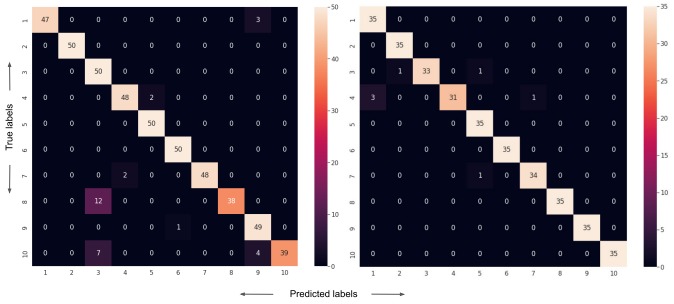


Fig. 11. Confusion matrix. Left: Ouhands. Right: NUS.

TABLE IX

COMPARISON OF THE PROPOSED CLASSIFICATION METHOD WITH THE STATE-OF-THE-ART METHODS FOR THE OUHANDS AND NUS DATASETS

Methods	Dataset	Accuracy (%)
Dadashzadeh *et al.* [1]	Ouhands	87.8
Matilainen *et al.* [8]	Ouhands	83.25
Bose *et al.* [42]	NUS	97.98
Aditya *et al.* [43]	NUS	94.7
Pisarady *et al.* [9]	NUS	94.36
Sharma *et al.* [44]	NUS	96.62
Bhaumik *et al.* [45]	Ouhands	65.1
Bhaumik *et al.* [46]	NUS	97.78
Sahoo *et al.* [47]	NUS	94.80
Ours	Ouhands	**93.8**
	NUS	**98.0**

TABLE X

STUDY OF THE PERFORMANCE OF FIVE USERS FOR FIVE ATTEMPTS ON PAS INTERFACE

Attempts	User 1	User 2	User 3	User 4	User 5
1	83.3	86.5	87.2	88.4	88.2
2	87.6	89.3	87.0	86.3	91.8
3	89.9	92.4	88.1	87.6	90.8
4	90.0	90.2	93,5	89.1	90.4
5	92.1	90.6	92.1	89.0	89.1

Table VI. The AKCAL encoder had the best performance with 93.8% accuracy.

Also, there is a hyperparameter ρ associated with the adaptive kernel size of AKCAL. We experimented with different values of ρ and found that $\rho = 2$ had the best result (shown in Table VII). The advantage of using weighting schemes is shown in Table VIII. The best performance is derived when both the average and max weighting schemes are used. The effects of using AKCAL are depicted in Fig. 10. AKCAL resulted in a packed heatmap instead of a dispersed one, as shown in Fig. 10 (bottom). The intuition behind using AKCAL is that channel attention focuses on what to attend in a feature map. We showed the heatmap of input images to illustrate what locations the proposed network focus on to learn classification-related features in the presence and absence of the AKCAL layer. The AKCAL helps the network focus on only the hand region and not capture features from the adjoining background region. This results in the compact heatmap. But, without the AKCAL layer, the resulting heatmap does not converge on the hand region, suggesting that noisy information is also encoded. This reduces the recognition accuracy.

The confusion matrix for the two datasets is shown in Fig. 11. Table IX shows comparison of some of the state-of-the-art methods with the proposed method, and it is observed that the proposed method performs better than the others. It further accentuates the robustness of the proposed method.

Moreover, five users were asked to operate the interface in the laboratory environment and test its reliability. Due consent from the users and the institute's ethics committee was taken for the experiments. The users were asked to keep their arms still while performing the gestures displayed on the interface five times. The mean accuracies of the gestures for each attempt are recorded in Table X. The accuracy ranged from 89.0% to 92.1%, which indicates the reliability and effectiveness of the PAS system. However, for failed cases, the users were asked to perform the gesture to call a nurse or a family member. Thus, the interface would present a way to establish communication even if a gesture was

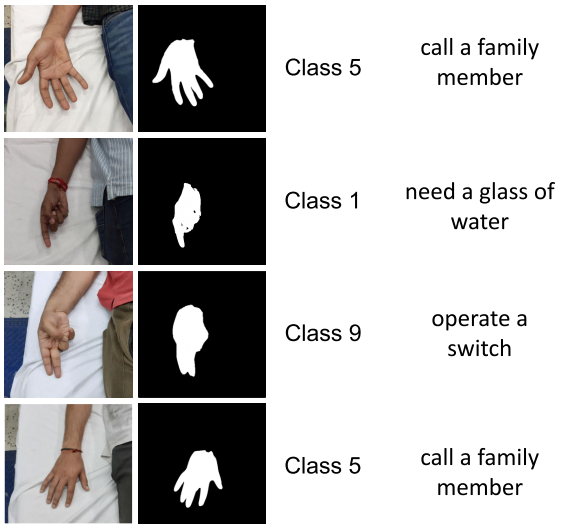


Fig. 12. PAS output for users performing the gestures in the laboratory environment. The first column shows the image frames captured by the setup shown in Fig. 2, the second column shows the predicted saliency map, the third column shows the classification results, and the fourth column shows the messages associated with the classes.

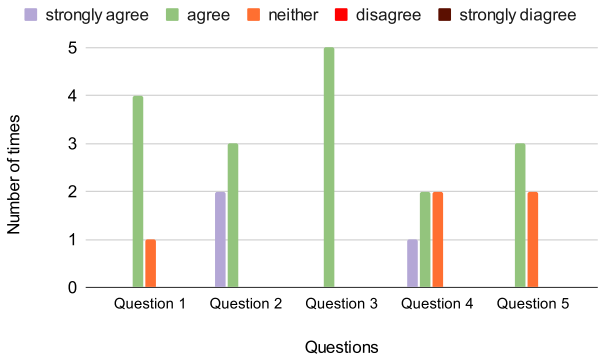


Fig. 13. Feedback response of the patients about the PAS.

misrecognized. This is facilitated by a confirmation message after each gesture, which displays "yes" and "no" options. The user must select the desired option to proceed. In this case, "yes" is selected by an open hand gesture (class 5) and "no" by a fist (class 10). We also took a few real-scene images of the users to verify our proposed approach. As shown in Fig. 12, the PAS associated the correct messages with the gestures posed by the users. It shows that the system is operable in a real scenario and performs reliably.

We further prepared the following questionnaire to obtain feedback about the usefulness of the PAS.

1) The proposed system is operable without help after the initial demonstration of the system.
2) The proposed system is useful and gives a satisfactory feel.
3) The proposed system will be used by the patients again.
4) The proposed system is useful for medical staff and reduced miscommunication (due to a medical staff attending to a request he cannot address).
5) The proposed system needs more functions and improvement.

The results are summarized in Fig. 13. For the most part, the patients found the system operable and useful, and they would consider using it again. They also pointed out that they would appreciate further advancement related to its speed of operation and increased functionality.

VI. Conclusion

A hand-gesture-controlled PAS is proposed, which uses a two-stage hand recognition architecture to combine the benefits of a convolution and transformer architecture. We developed a novel saliency detection method that largely overcomes the challenges posed by vision-based hand gesture recognition, such as occlusion, background clutter, varying illumination, and the presence of skin regions. The saliency map thus obtained highlighted the hand region and was fed to the classification network. This network used an AKCAL that emphasized the features relevant for classification. The recognition accuracy for the two benchmark datasets was 93.8% and 98.0%, which highlights the preciseness of the proposed approach. The PAS uses this recognition approach and assists the patients in communicating their needs to the medical staff. The patient can send ten different messages using this system, which the call button commonly found in hospitals cannot.

Limitations and Future Work: The proposed system is a two-stage system, making the second stage dependent on the first stage for the best output. Therefore, it can achieve inference time suitable for online applications (<25 or 30 frames/s), not real-time (=25 or 30 frames/s). In the future, we will develop a single-stage architecture that can achieve real-time performance. The system tends to underperform if many skin objects are available in the background (like multiple faces or hands). However, the patient's hand is the only skin object the system is likely to capture in a hospital environment. Nevertheless, we will work on achieving better performance in the presence of multiple skin objects. Moreover, the PAS could incorporate more messages to improve communication between the patient and medical staff further. Also, it could incorporate automatic dizziness detection and alert the medical staff if the patient lapsed into unconsciousness.

References

[1] A. Dadashzadeh, A. T. Targhi, M. Tahmasbi, and M. Mirmehdi, "HGR-Net: A fusion network for hand gesture segmentation and recognition," *IET Comput. Vis.*, vol. 13, no. 8, pp. 700–707, Dec. 2019, doi: 10.1049/iet-cvi.2018.5796.

[2] N. Adaloglou et al., "A comprehensive study on deep learning-based methods for sign language recognition," *IEEE Trans. Multimedia*, vol. 24, pp. 1750–1762, 2022, doi: 10.1109/TMM.2021.3070438.

[3] G. Plouffe and A. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, Feb. 2016, doi: 10.1109/TIM.2015.2498560.

[4] J. Nalepa and M. Kawulok, "Fast and accurate hand shape classification," in *Beyond Databases, Architectures, and Structures* (Communications in Computer and Information Science), vol. 424, S. Kozielski, D. Mrozek, P. Kasprowski, B. Malysiak-Mrozek, and D. Kostrzewa, Eds. Cham, Switzerland: Springer, 2014, pp. 364–373, doi: 10.1007/978-3-319-06932-6_35.

[5] T. Grzejszczak, M. Kawulok, and A. Galuszka, "Hand landmarks detection and localization in color images," *Multimedia Tools Appl.*, vol. 75, no. 23, pp. 16363–16387, Dec. 2016, doi: 10.1007/s11042-015-2934-5.

[6] X. Zhang et al., "Hand image understanding via deep multi-task learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11261–11272, doi: 10.1109/ICCV48922.2021.01109.

[7] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1949–1957, doi: 10.1109/ICCV.2015.226.

[8] M. Matilainen, P. Sangi, J. Holappa, and O. Silvén, "OUHANDS database for hand detection and pose recognition," in *Proc. 6th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Dec. 2016, pp. 1–5, doi: 10.1109/IPTA.2016.7821025.

[9] P. K. Pisharady, P. Vadakkepat, and A. P. Loh, "Attention based detection and recognition of hand postures against complex backgrounds," *Int. J. Comput. Vis.*, vol. 101, no. 3, pp. 403–419, Feb. 2013, doi: 10.1007/s11263-012-0560-5.

[10] M. Kawulok, J. Kawulok, and J. Nalepa, "Spatial-based skin detection using discriminative skin-presence features," *Pattern Recognit. Lett.*, vol. 41, pp. 3–13, May 2014, doi: 10.1016/j.patrec.2013.08.028.

[11] R. Khan, A. Hanbury, J. Stöttinger, and A. Bais, "Color based skin classification," *Pattern Recognit. Lett.*, vol. 33, no. 2, pp. 157–163, Jan. 2012, doi: 10.1016/j.patrec.2011.09.032.

[12] B. K. Chakraborty and M. K. Bhuyan, "Image specific discriminative feature extraction for skin segmentation," *Multimedia Tools Appl.*, vol. 79, nos. 27–28, pp. 18981–19004, Jul. 2020, doi: 10.1007/s11042-020-08762-4.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.

[14] O. Oktay et al., "Attention U-Net: Learning where to look for the pancreas," 2018, *arXiv:1804.03999*.

[15] J. Chen et al., "TransUNet: Transformers make strong encoders for medical image segmentation," 2021, *arXiv:2102.04306*.

[16] A. U. Khan and A. Borji, "Analysis of hand segmentation in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4710–4719, doi: 10.1109/cvpr.2018.00495.

[17] M. Cai, F. Lu, and Y. Sato, "Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14380–14389, doi: 10.1109/cvpr42600.2020.01440.

[18] F. Yang and Y. Wu, "A soft proposal segmentation network (SPS-Net) for hand segmentation on depth videos," *IEEE Access*, vol. 7, pp. 29655–29661, 2019, doi: 10.1109/ACCESS.2019.2900991.

[19] H. P. Jyoti Dutta, D. Sarma, M. K. Bhuyan, and R. H. Laskar, "Semantic segmentation based hand gesture recognition using deep neural networks," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6, doi: 10.1109/NCC48643.2020.9055990.

[20] P. Bao, A. I. Maqueda, C. R. del-Blanco, and N. García, "Tiny hand gesture recognition without localization via a deep convolutional network," *IEEE Trans. Consum. Electron.*, vol. 63, no. 3, pp. 251–257, Aug. 2017, doi: 10.1109/TCE.2017.014971.

[21] S. F. Chevtchenko, R. F. Vale, V. Macario, and F. R. Cordeiro, "A convolutional neural network with feature fusion for real-time hand posture recognition," *Appl. Soft Comput.*, vol. 73, pp. 748–766, Dec. 2018, doi: 10.1016/j.asoc.2018.09.010.

[22] T. H. N. Le, K. G. Quach, C. Zhu, C. N. Duong, K. Luu, and M. Savvides, "Robust hand detection and classification in vehicles and in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1203–1210, doi: 10.1109/CVPRW.2017.159.

[23] N. Park and S. Kim, "How do vision transformers work?" in *Proc. Int. Conf. Learn. Represent.*, 2022.

[24] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Red Hook, NY, USA: Curran Associates, 2017, pp. 6000–6010.

[25] S. Zhuoran, Z. Mingyuan, Z. Haiyu, Y. Shuai, and L. Hongsheng, "Efficient attention: Attention with linear complexities," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3530–3538, doi: 10.1109/WACV48630.2021.00357.

[26] J. Lei Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.

[27] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007, doi: 10.1109/ICCV.2017.324.

[28] S. Shit et al., "ClDice—A novel topology-preserving loss function for tubular structure segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 16555–16564, doi: 10.1109/cvpr46437.2021.01629.

[29] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440, doi: 10.1109/CVPR.2015.7298965.

[30] Q. Jin, Z. Meng, C. Sun, H. Cui, and R. Su, "RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans," *Frontiers Bioeng. Biotechnol.*, vol. 8, p. 1471, Dec. 2020.

[31] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 833–851.

[33] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.

[34] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149, doi: 10.1109/cvpr.2019.00326.

[35] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3138–3147, doi: 10.1109/WACV48630.2021.00318.

[36] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/f2201f5191c4e92cc5af043eebfd0946-Paper.pdf

[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239, doi: 10.1109/CVPR.2017.660.

[38] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Computer Vision—ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 418–434.

[39] J. Wang et al., "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3349–3364, Oct. 2021, doi: 10.1109/TPAMI.2020.2983686.

[40] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds., May 2015.

[42] S. R. Bose and V. S. Kumar, "Efficient inception v2 based deep convolutional neural network for real-time hand action recognition," *IET Image Process.*, vol. 14, no. 4, pp. 688–696, Mar. 2020, doi: 10.1049/iet-ipr.2019.0985.

[43] V. Adithya and R. Rajesh, "A deep convolutional neural network approach for static hand gesture recognition," *Proc. Comput. Sci.*, vol. 171, pp. 2353–2361, Jan. 2020, doi: 10.1016/j.procs.2020.04.255.

[44] S. Sharma, H. Pallab Jyoti Dutta, M. K. Bhuyan, and R. H. Laskar, "Hand gesture localization and classification by deep neural network for online text entry," in *Proc. IEEE Appl. Signal Process. Conf. (ASPCON)*, Oct. 2020, pp. 298–302, doi: 10.1109/ASPCON49795.2020.9276713.

[45] G. Bhaumik, M. Verma, M. C. Govil, and S. K. Vipparthi, "ExtriDeNet: An intensive feature extrication deep network for hand gesture recognition," *Vis. Comput.*, vol. 38, no. 11, pp. 3853–3866, Nov. 2022, doi: 10.1007/s00371-021-02225-z.

[46] G. Bhaumik, M. Verma, M. C. Govil, and S. K. Vipparthi, "HyFiNet: Hybrid feature attention network for hand gesture recognition," *Multimedia Tools Appl.*, vol. 82, no. 4, pp. 4863–4882, Feb. 2023, doi: 10.1007/s11042-021-11623-3.

[47] J. Sahoo, S. Sahoo, S. Ari, and S. Patra, "RBI-2RCNN: Residual block intensity feature using a two-stage residual convolutional neural network for static hand gesture recognition," *Signal, Image Video Process.*, vol. 16, pp. 2019–2027, Feb. 2022, doi: 10.1007/s11760-022-02163-w.

H. Pallab Jyoti Dutta received the M.Tech. degree from the Department of Electronics and Communication Engineering, Gauhati University, Guwahati, India, in 2018. He is currently pursuing the Ph.D. degree with the Department of Electronics and Electrical Engineering, Indian Institute of Technology (IIT) Guwahati, Guwahati.

His research interests include deep learning, human–computer interaction (HCI), image processing, and machine learning.

M. K. Bhuyan (Senior Member, IEEE) received the Ph.D. degree in electronics and communication engineering from IIT Guwahati, Guwahati, India, in 2005.

He is a Professor with the Department of Electronics and Electrical Engineering, IIT Guwahati. His current research interests include artificial intelligence, augmented and virtual reality, biomedical signal processing, computer vision, and human–computer interaction (HCI).

Dr. Bhuyan was a recipient of the National Award for Best Applied Research/Technological Innovation.

Debanga Raj Neog received the Ph.D. degree in computer science from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2018.

He is currently an Assistant Professor with the Mehta Family School of Data Science and Artificial Intelligence, IIT Guwahati, Guwahati, India. He also co-founded Nytilus Inc., Toronto, ON, Canada. His research interests include augmented and virtual reality, computational imaging, computer graphics, and computer vision.

Dr. Neog was a recipient of the Mitacs Globalink Research Award.

Karl Fredric MacDorman received the Ph.D. degree in computer science from Cambridge University, Cambridge, U.K, in 1997.

He is an Associate Professor in the Human–Computer Interaction Program with the Luddy School of Informatics, Computing, and Engineering, Indiana University, Indianapolis, IN, USA. He is also the Director of the Informatics and Artificial Intelligence Programs and the Associate Dean of Academic Affairs. His research interests include cognitive science, human–computer interaction (HCI), machine learning, and robotics.

Rabul Hussain Laskar received the Ph.D. degree from the National Institute of Technology (NIT), Silchar, Silchar, India, in 2012.

He is an Associate Professor with the Department of Electronics and Communication Engineering, NIT, Silchar. His research interests include communication engineering, image processing, soft computing techniques, and speech processing.