Expert Systems With Applications 234 (2023) 121046

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Efficient hand segmentation for rehabilitation tasks using a convolution neural network with attention

H Pallab Jyoti Dutta [a,*], M.K. Bhuyan [a,b], Debanga Raj Neog [b], Karl Fredric MacDorman [c], Rabul Hussain Laskar [d]

[a] Department of Electronics and Electrical Engineering, Indian Institute of Technology, Guwahati, Assam, 781039, India
[b] Mehta Family School of Data Science and Artificial Intelligence, Indian Institute of Technology, Guwahati, Assam, 781039, India
[c] Luddy School of Informatics, Computing, and Engineering, Indiana University, Indianapolis, IN, 46202, USA
[d] Department of Electronics and Communication Engineering, National Institute of Technology, Silchar, Assam, 788010, India

ARTICLE INFO

Keywords:
Channel attention
Efficient attention mechanism
Encoder–decoder
Hand rehabilitation
Hand segmentation
Spatial attention

ABSTRACT

We designed an interface to support hand rehabilitation tasks to restore hand function and relieve discomfort. The interface requires accurate hand segmentation, which is impeded by background clutter, occlusion, and variations in illumination. To overcome these challenges, we propose a novel encoder–decoder that segments the hand by encoding spatial and channel correlations using two attention blocks. This approach requires much less computation than benchmark self-attention mechanisms. Moreover, a novel loss function optimizes the model to resolve class imbalance, ensure boundary smoothness, and retain the hand's shape. The quantitative and qualitative results show the model's ability to segment the hands. It performed exceptionally well for images with different hand poses and orientations, the presence of a human face, background clutter, specularity, and variations in illumination. The model attained an F1-score of 97.3% for the Ouhands and 99.3% for the HGR dataset, higher than baseline models, with faster inference times. Furthermore, the model could generalize hand segmentation to multiple hands and unseen environments. Its segmentation precision enabled the development of the hand rehabilitation interface, which guided users to perform hand exercises. For five weeks, patients steadily improved hand function while using the interface.

1. Introduction

Hands play an essential role in daily activities. They support communication among people, especially those who are deaf or mute and rely on sign language (Adaloglou et al., 2022; Chakraborty et al., 2018; Mitra & Acharya, 2007). For this community and the general population, gesture recognition is finding many applications. It lets people operate contactless interfaces for teleconferencing, presentations, home appliances, robots, and drones. However, tremors, stiffness, and spasticity resulting from injuries, nerve and muscle complications, and medical conditions can impair movement, making it hard to use the hands to interact or perform everyday tasks.

Physical therapy is often effective in restoring hand function (Luo et al., 2010). An interface for teaching patients simple hand exercises and guiding their progress should help improve dexterity. However, such an interface requires the detection of hand movements. This information can be acquired using wearable devices like data gloves or electromyography (Mitra & Acharya, 2007). Unfortunately, these devices can restrict movement, cause fatigue, and increase pain. Unlike video cameras, they are not commonly available. Thus, this study uses a vision-based method to localize hand movements and segment the hand region. However, the performance of vision-based methods degrades under such real-world conditions as background clutter, occlusion, variations in illumination, and overlapping skin regions like the hands and face (Chakraborty et al., 2018). This work develops a novel architecture to overcome these barriers and segment the hand region from the background.

The architecture is incorporated into a human–computer interface for physical therapy. This interface helps improve hand and finger mobility by prompting patients to make hand gestures to grab an object and drag it to a specific location. In addition, the interface prompts them to traverse a path while remaining within a boundary. These exercises help stabilize hand movements. Thus, localizing the hand is the prime objective and it requires accurate hand segmentation. A schematic diagram describing the entire process is shown in Fig. 1.

* Corresponding author.
E-mail addresses: h18@iitg.ac.in (H.P.J. Dutta), mkb@iitg.ac.in (M.K. Bhuyan), dneog@iitg.ac.in (D.R. Neog), kmacdorm@indiana.edu (K.F. MacDorman), rhlaskar@ece.nits.ac.in (R.H. Laskar).

https://doi.org/10.1016/j.eswa.2023.121046
Received 23 January 2023; Received in revised form 7 July 2023; Accepted 22 July 2023
Available online 29 July 2023

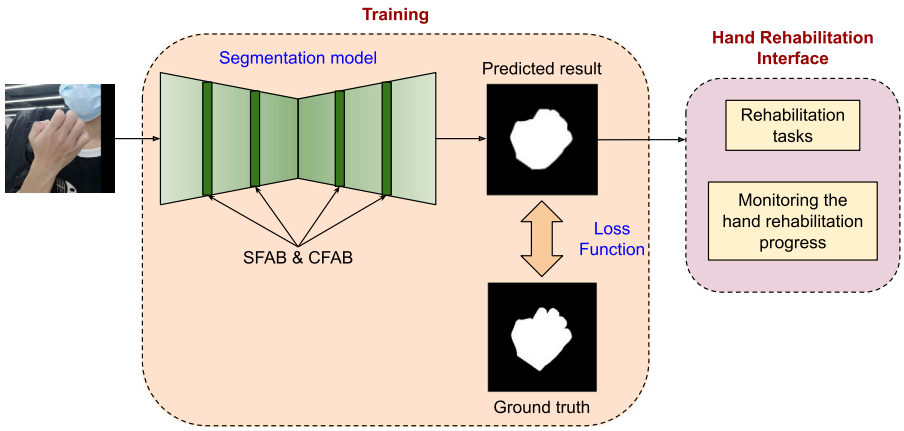


Fig. 1. The schematic representation of the proposed method.

The input image is passed through the segmentation model to obtain the hand mask. Then, the mask is used to perform rehabilitation tasks prompted by the interface. Finally, the progress of the patient undergoing the rehabilitation program is monitored for a period of time, and the usefulness of the hand rehabilitation interface is analyzed.

The proposed segmentation model is inspired by the DANet (Fu et al., 2019), which uses attention-based spatial and channel attention for better semantic segmentation. However, the attention modules used in DANet are memory-intensive. We propose attention modules requiring much less memory that enhance features belonging to the hand region and help obtain an accurate hand-segmented image. Our segmentation model is an encoder–decoder network with two unique attention blocks. The first generates attention maps for the spatial dimension to locate the region of interest. The second creates attention maps for the channel dimension to assign the region to the foreground (i.e., the hand) or background, focusing on class-specific attributes. Using an efficient self-attention mechanism (SAM), the spatial feature attention block (SFAB) encodes the correlation between one position of a particular channel of the feature map and the remaining positions. The channel feature attention block (CFAB) correlates a position of the feature map across the channels using a similar efficient self-attention mechanism. The input image passes through two blocks in the encoder, a sequential arrangement of SFAB, convolution block, max-pooling layer, and CFAB and two blocks in the decoder, a sequential arrangement of convolution block, SFAB, upsampling layer, convolution block, and CFAB. It also incorporates skip connections from the encoder to the decoder to pass on fine details and avoid vanishing gradients. Subsequent sections detail the network's design and analysis, including variations on its structure.

This work makes the following contributions:

1. We propose a novel architecture for hand segmentation that uses an attention mechanism to capture long-range (global) characteristics and obtain accurate hand masks.
2. The architecture has two novel and efficient attention blocks, one for spatial features to emphasize the hand's location and one for channel features to emphasize the pixel class.
3. A composite loss function is proposed to address class imbalance, ensure boundary smoothness, and retain the geometric shape of the hand, especially around the fingers.
4. We developed an interface that guides patients with hand or arm injuries or a lack of dexterity through hand rehabilitation tasks. The tasks were designed to help them regain normal hand function by targeting different muscles and motor nerves. The results show that the interface significantly improves performance.
5. Our method achieves state-of-the-art performance, both for accuracy and computational complexity, on benchmark single-hand datasets, such as Ouhands and HGR. Moreover, as shown on the NUS II dataset and images containing two hands, it can generalize.

The paper is arranged into five sections. Section 2 highlights previous work on segmentation. Section 3 describes the proposed architecture, including the SFAB, CFAB, and composite loss function. Section 4 explains the hand rehabilitation tasks. Section 5 presents the experiments and the quantitative and qualitative results. Section 6 concludes the paper.

2. Related works

In computer vision and human–robot interaction (Ju et al., 2017), hand segmentation finds a wide range of applications, requiring hand pose estimation (Wang et al., 2019) and hand gesture recognition (Dadashzadeh et al., 2019). It aims to label hand region pixels as foreground and other pixels as background. Much research has been performed using designer-specific features. Kawulok et al. (2014a) used spatial and texture-based features to model the skin region. They generated skin probability maps and applied linear discriminant analysis to obtain features discriminating the skin areas, subsequently detecting the skin regions upon spatial analysis. Khan et al. (2012) analyzed the effect of color space and illumination on accurately detecting skin regions. Moreover, they studied the performance of nine skin modeling algorithms and found that the color constancy algorithm improves skin detection. An unsupervised approach is adopted by Chakraborty and Bhuyan (2020) to obtain features that differentiate an image's skin and non-skin areas using adaptive discriminative analysis. Although color and skin-based detection worked for hand segmentation, the spotlight shifted to deep learning because it promised better and human-level detection without needing designer-specified features.

In Dadashzadeh et al. (2019), residual blocks and atrous spatial pyramid pooling blocks produced segmented hand masks, enabling a classification network to recognize hand gestures. Khan and Borji (2018) performed hand segmentation in egocentric videos using RefineNet, a deep learning model, and improved the results by employing conditional random fields. Cai et al. (2020) used a Bayesian convolution neural network (CNN) to estimate model uncertainty and share hand shape information across different domains to enable generalization in hand segmentation. In Dutta et al. (2020), a UNet architecture achieved the best hand segmentation results for sign language recognition. Yang and Wu (2019) proposed SPSNet, a novel architecture for hand segmentation that fuses temporal and tracking proposals in depth videos and reduces the complexity of hand pose estimation. Wang et al. (2019) proposed a two-stage CNN. The first stage generated a hand mask while the second estimated hand joints. Tsai and Huang (2022) proposed a two-stage segmentation network, where the first stage produces a rough segmentation mask, which is improved by the second stage employing

a distance method. The first stage is a simplified U-Net architecture followed by a refined block in the second stage, and the authors named the complete architecture Refined Simple U-Net. In Ohkawa et al. (2021), the authors created a source dataset that has been style-adapted to look like the target dataset. Then, a segmentation model is trained on the style-adapted dataset and a reference model on the original dataset. The intersection of the masks generated by the two networks produces the pseudo labels. These pseudo labels are used to update the segmentation model again, which eventually generates the required segmentation hand masks.

Methods from related research areas can be adapted to hand segmentation. Wu et al. (2019) proposed a joint pyramid upsampling technique for dilated convolution to reduce its complexity and memory requirements for semantic image segmentation. Fu et al. (2019) proposed DANet, which uses two self-attention mechanisms to capture a scene's contextual information. They encode position and channel information, yielding a feature representation for scene segmentation. Ronneberger et al. (2015) developed U-Net, an architecture to segment medical images with vastly fewer training samples. It consists of a contracting path that encodes context and an expanding path that localizes the object. An extension, Attention U-Net, was proposed by Oktay et al. (2018). Its attention gating mechanism focuses on the object of interest, ignoring irrelevant areas. This mechanism was modified by Jin et al. (2020) to include residual learning. Although these three U-Net architectures were designed for medical images, they have been adapted to semantic segmentation because of their exceptional performance. Sun et al. (2021) used Gaussian dynamic convolution with a dynamic receptive field for fast single image segmentation. Baheti et al. (2020) proposed an encoder–decoder architecture, named Eff-UNet, for semantic segmentation where the encoder is an EfficientNet and the decoder is taken from UNet. In Almeida et al. (2021), Mask-RCNN was used to segment the hands of a humanoid robot in ego-centric images.

Despite these contributions, several challenges remain for hand segmentation. These include generalizing segmentation to novel images and maintaining performance given occlusion, exposed skin areas other than the hand, and variations in illumination. Most works generate segmentation masks at the expense of memory complexity (more training parameters), involving two-stage computation or post-processing. Also, the models designed for semantic segmentation do not appropriately consider the contours of the hand, especially near the fingers, resulting in a coarse segmentation mask. Hence, we sought to close this research gap by developing a novel, efficient, and effective solution.

3. Methodology

This section describes the architecture that obtains the segmented hand region. Also, it discusses the importance of the SFAB and CFAB modules.

3.1. Overview

Segmentation can be modeled as a pixel-wise binary classification task with at least two classes—foreground and background. A deep neural network uses convolution layers to encode the features of the two classes. In a CNN, each receptive field of the convolution layers is sensitive to local features in the image. However, it becomes harder for a CNN to detect patterns at increasing spatial and temporal scales. With each layer, within-class pattern differences reduce the network's recognition accuracy. To overcome this, information from distal pixels of the same class must be integrated. Therefore, we propose an architecture that encodes distal pixel dependencies using a self-attention mechanism for spatial and channel dimensions, as shown in Fig. 2.

The spatial feature attention block implements spatial self-attention, and the channel feature attention block implements channel self-attention. SFAB integrates the spatial correlations of different pixels to obtain spatial attention maps that focus on regions likely to contain the object. CFAB complements SFAB by generating attention maps that focus on the object's class, capturing correlations among the channels of the input feature map. The proposed architecture is an encoder–decoder network with the encoder containing four convolution blocks, two SFAB and CFAB blocks, and a depthwise convolution layer through which the output of the encoder is propagated to the decoder. The decoder also contains four convolution blocks and two SFAB and CFAB blocks. The output of the decoder passes through a sequential arrangement of a convolution layer with a 1×1 filter size, a batch normalization layer, and sigmoid activation to obtain the segmented mask.

The encoder's sequential ordering is SFAB–convolution block–max-pooling–CFAB. The intuition behind this ordering is that the SFAB attends to the object's regions, and this feature map is passed to the convolution block. The convolution block thus learns the features belonging to the foreground region, namely, those of the hand. A max-pooling layer is placed after the convolution block, followed by a CFAB. This layer captures distinctive features of each class to improve channel-specific attention (Woo et al., 2018). A CFAB follows to capture channel-specific details and accentuates class-dependent features. The experimental analysis of this arrangement is detailed in section IV-B. A pair of convolution and batch normalization layers comprise the convolution block, as shown in Fig. 2. Each convolution layer has a 3×3 filter size. The batch normalization layer standardizes the input feature map to reduce training time.

The input image propagates through a convolution block and a max-pooling layer to the first SFAB–convolution block–max-pooling–CFAB sequence. The resulting attention feature maps pass through another convolution block and max-pooling layer, followed by the second SFAB–convolution block–max-pooling–CFAB sequence. Before passing the resultant feature maps to the decoder, a depthwise convolution layer encodes it. The advantage of a depthwise convolution layer over a standard convolution layer is that it introduces much fewer training parameters to the deep neural network. The four max-pooling layers reduce the spatial dimension of the encoder's output feature maps to $\frac{1}{16}$th the input image. Therefore, the decoder upsamples the feature maps by a factor of 2 and concatenates the upsampled feature maps with the output of the previous convolution block, i.e., Conv Block 4. This skip connection incorporates details from the encoder layers into the decoder layers to arrive at a more refined segmentation mask at the decoder's output. The concatenated output is passed through a convolution block–SFAB–upsampling layer sequence. The resulting feature map is added with the output of Conv Block 3 and passed to a convolution block–CFAB–upsampling layer sequence. The output is then concatenated with Conv Block 2's output, passed through a convolution block–SFAB–upsampling layer sequence, and subsequently concatenated with Conv Block 1's output. Next, the propagated feature map is passed through the final convolution block, the CFAB, a sequence of convolution layer of filter size 1×1, a batch normalization layer, and a sigmoid activation layer to arrive at the final output, that is, the segmentation mask.

3.2. Self-attention mechanism

The self-attention mechanism models long-range dependencies (Vaswani et al., 2017). Initially designed for natural language processing, SAMs have been used extensively in computer vision to capture an image's global context (Fu et al., 2019; Zhuoran et al., 2021). The SAM incorporates the global and local context by using convolution. A pictorial representation of a SAM is shown in Fig. 3.

A feature map $F \in \mathbb{R}^{h \times w \times c}$ is reshaped into $X \in \mathbb{R}^{hw \times c}$ and passed through the SAM to obtain the output attention map. Here, h, w, and c represent the feature map's height, width, and channel. For the mth position of X, $\exists$ a vector $\mathbf{x}_m \in \mathbb{R}^c$. Then, three vectors, namely, the query vector $\mathbf{x}_{qm} \in \mathbb{R}^{d_k}$, key vector $\mathbf{x}_{km} \in \mathbb{R}^{d_k}$, and value vector $\mathbf{x}_{vm} \in \mathbb{R}^{d_v}$, are constructed from $\mathbf{x}_m$. The SAM estimates the relevance of the

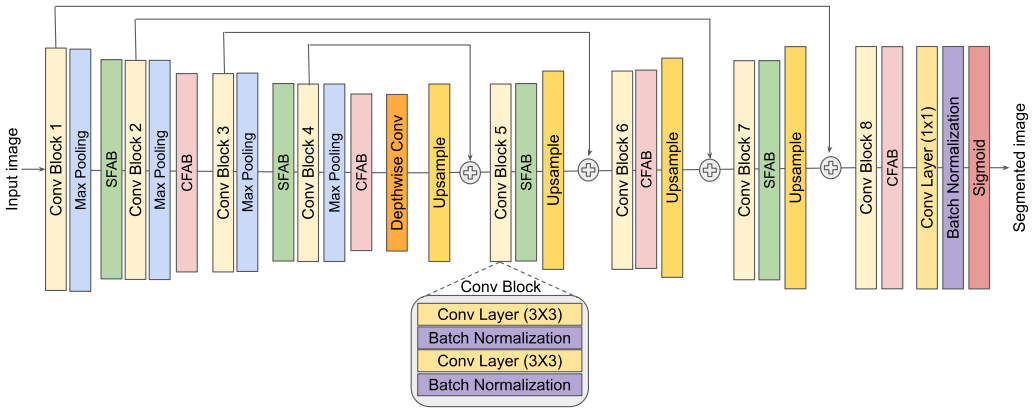


Fig. 2. The block diagram of the proposed segmentation method.

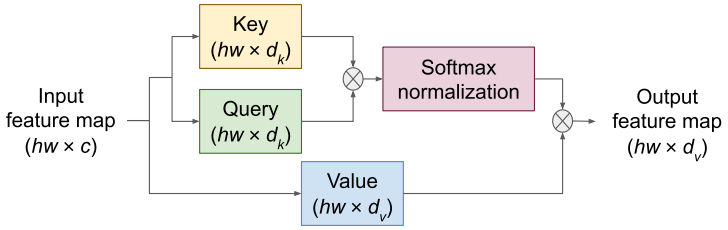


Fig. 3. Block representation of the self-attention mechanism.

nth position to the mth position, i.e., the current position, to improve the current feature vector's encoding. Thus, the correlation between the mth query vector and the nth key vector is obtained, i.e., $\mathbf{x}_{qm}{}^{T}\mathbf{x}_{kn}$ where $n \in 1, 2, \ldots, hw$. This dot product results in a score, which is divided by $\sqrt{d_k}$ to stabilize the gradients and then normalized by the softmax function to ensure positive scores. This score is represented as

$$score_{mn} = \frac{exp(\frac{\mathbf{x}_{qm}^{T}\mathbf{x}_{kn}}{\sqrt{d_k}})}{\sum_{j=1}^{hw} exp(\frac{\mathbf{x}_{qm}^{T}\mathbf{x}_{kj}}{\sqrt{d_k}})}. \tag{1}$$

It is the correlation between the position pairs m and n. The score is multiplied with the mth value vector $\mathbf{x}_{vm}$ to obtain the weighted value vector at the mth position. The weighted value vectors for other positions are obtained in the same way. Relevant positions are multiplied by a large score and are thus brought into focus, while irrelevant positions are multiplied by a small score and are thus ignored. Finally, the weighted value vectors are added to obtain a vector $\mathbf{x}_{\text{sum}_m}$, the SAM's output for the mth position, which is expressed as

$$\mathbf{x}_{\text{sum}_m} = \sum_{j=1}^{hw} score_{mj} \times \mathbf{x}_{vj}. \tag{2}$$

For all the hw positions, we create the query matrix $\mathbf{M}_q \in \mathbb{R}^{hw \times d_k}$, key matrix $\mathbf{M}_k \in \mathbb{R}^{hw \times d_k}$, and value matrix $\mathbf{M}_v \in \mathbb{R}^{hw \times d_v}$. Thus, the SAM's attention map is given by

$$\text{Output feature map} = softmax\left(\frac{\mathbf{M}_q\mathbf{M}_k^{T}}{\sqrt{d_k}}\right)\mathbf{M}_v. \tag{3}$$

For self-attention $d_k = d_v$.

3.3. SFAB and CFAB

Although the SAM's ability to encode long-range dependencies, its memory usage and high computational complexity are drawbacks. Determining position-to-position correlation is $\mathcal{O}((hw)^2)$ for memory and $\mathcal{O}(d_k(hw)^2)$ for computation. Under memory constraints, the SAM can cause an out-of-memory error. Therefore, drawing inspiration from Zhuoran et al. (2021), we propose two novel efficient attention blocks that generate attention maps along spatial and channel dimensions. These blocks, namely SFAB and CFAB, reduce memory use from quadratic to linear.

In this case, the queries $\mathbf{M}_q$, keys $\mathbf{M}_k$, and values $\mathbf{M}_v$ also exist. However, instead of calculating the correlations between positions, each channel of keys is considered a feature map and is multiplied by the values. This results in a global feature map weighting every position of the input feature map and highlighting its class-specific features. Now the queries at each position aggregate the global feature maps and generate the final attention map that attends to the object's class and position.

The SFAB uses this efficient way of calculating attention along the spatial dimension to focus on the location of the foreground hand region. At the outset, it arranges the queries, keys, and values into groups to introduce a parallel and independent way of attending to different representations of the feature map. Thus, there are $(\#channels/\#divisions)$ groups. The queries, keys, and values of the first group are represented as $\mathbf{M}_q^1$, $\mathbf{M}_k^1$, $\mathbf{M}_v^1 \in \mathbb{R}^{hw \times \frac{d_k}{\#divisions}}$, respectively. $\#channels$ and $\#divisions$ represent the number of channels and divisions, respectively. The associative property of matrix multiplication ensures $(\mathbf{M}_q\mathbf{M}_k^{T})\mathbf{M}_v = \mathbf{M}_q(\mathbf{M}_k^{T}\mathbf{M}_v)$. However, Zhuoran et al. (2021) state that, to realize this matrix associativity with the softmax function, two softmax functions are used—one for the queries and the other for the keys. These two softmax functions resemble the single softmax function of the SAM and approximate the required normalization. Thus, the attention expression for the first group is given by

$$attention^1 = softmax_q(\mathbf{M}_q^1)(softmax_k(\mathbf{M}_k^1)^{T}\mathbf{M}_v^1). \tag{4}$$

Similarly, $attention^1, \ldots, attention^{\#divisions}$ are appended together and fed to a convolution layer of filter size 1×1 to obtain the final spatial attention map of dimension $hw \times c$, as shown in Eq. (5). The SFAB is illustrated in Fig. 4.

$$\text{SFAB attention map} = conv([attention^1, attention^2, \ldots, attention^{\#divisions}]). \tag{5}$$

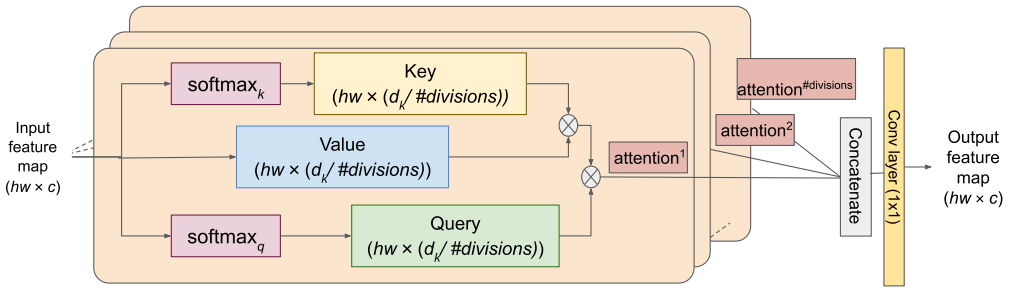


Fig. 4. Block diagram of the spatial feature attention block (SFAB).

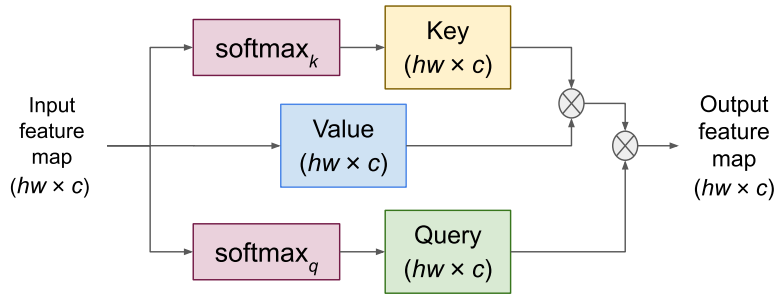


Fig. 5. Block diagram of the channel feature attention block (CFAB).

The CFAB has matrix multiplications like the SFAB but does not have divisions along channel dimensions to form groups and uses the entire key, query, and value matrices. Here, the channel relationship is encoded as attention maps, highlighting the semantics to segment the foreground hand from the background. Alternatively, it highlights the correlation between the mth and nth channels to attend to class-specific features. The CFAB is illustrated in Fig. 5, and the final channel attention map is shown in Eq. (6).

CFAB attention map =

$$softmax_q(\mathbf{M}_q)(softmax_k(\mathbf{M}_k)^T\mathbf{M}_v), \tag{6}$$

where the attention map is of dimension $hw \times c$ and $\mathbf{M}_q$, $\mathbf{M}_k$, $\mathbf{M}_v \in \mathbb{R}^{hw \times c}$, respectively.

This calculation method reduces memory complexity from $\mathcal{O}((hw)^2)$ to $\mathcal{O}(d_k hw + d_k^2)$ and computational complexity from $\mathcal{O}(d_k(hw)^2)$ to $\mathcal{O}((d_k)^2 hw)$. This avoids a quadratic increase in complexity with the feature map's spatial dimensions, and lets the developer set the value of c.

3.4. Loss function

During the training of a deep learning model, we minimize a loss function to learn the model's optimal weights for a given task. Therefore, we propose a novel composite loss function that integrates hand regions, segments the hand's shape, ensures boundary smoothness and continuity, and corrects for class imbalance. The loss function has these three components:

3.4.1. Binary focal loss

Segmentation is a pixel-wise classification task assigning pixels to the foreground or background. For hand segmentation, the foreground, i.e., the hand, has fewer pixels than the background because it covers a small area of the image. Hence, their proportion is imbalanced. Although binary cross entropy (BCE) has been used for segmentation, it is biased towards the background class because it has more pixels. Therefore, we used binary focal loss (BFL) (Lin et al., 2017). BFL reduces the contribution of background pixels that are easy to classify

Fig. 6. An image showing the finger valley and the slender part of the hand (indicated by cyan arrows →←).

while balancing the contribution of foreground and background pixels that are hard to classify. BFL develops on BCE, which is given by

$$BCE(\hat{y}, y) = \begin{cases} -log(\hat{y}), & y = 1 \\ -log(1-\hat{y}), & otherwise, \end{cases} \tag{7}$$

where $\hat{y}$ denotes the predicted probability and y denotes the true label. BFL uses a modulating factor β to reduce the contribution of easy-to-classify pixels to the loss function and amplify the contribution of hard-to-classify pixels, which would otherwise be low. The balancing factor α corrects for the imbalance in the foreground and background pixels. Thus, the BFL is defined as

$$BFL(\hat{y}, y) = \begin{cases} -\alpha(1-\hat{y})^\beta log(\hat{y}), & y = 1 \\ -(1-\alpha)(\hat{y})^\beta log(1-\hat{y}), & otherwise. \end{cases} \tag{8}$$

As the probability of correct prediction tends to 1, the scaling factor tends to 0, making the function resistant to class imbalance.

3.4.2. Smoothing loss

The hand is a deformable object, and the hand region's boundary in an image plays a pivotal role in the hand's accurate segmentation. The

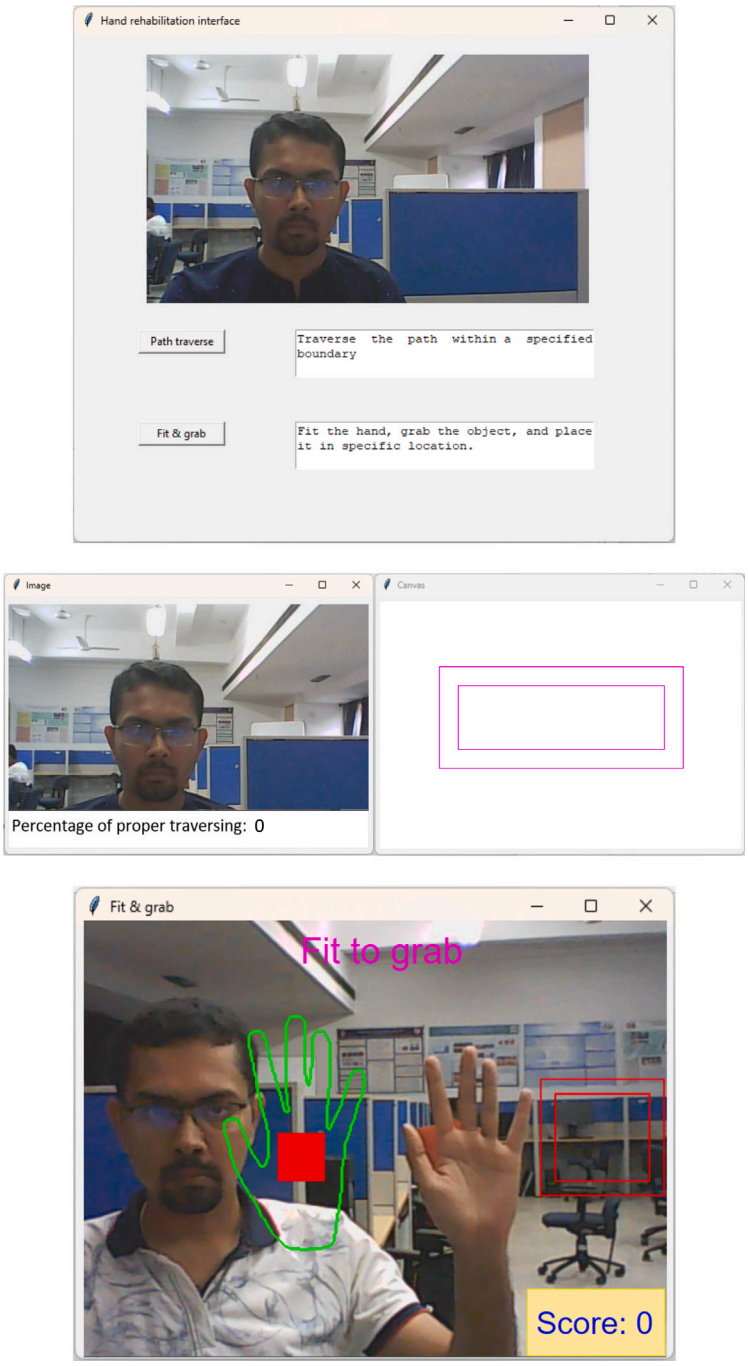


Fig. 7. The interface layout. Top: the main window. Middle: resulting window on clicking "Path traverse" button on the main window. Bottom: resulting window on clicking "Fit & grab" button on the main window.

fingers need attention to obtain an accurate segmentation mask because of their slender shape and constricted regions relative to the palm. For instance, the segmentation mask may not be smooth around finger valleys, as shown in Fig. 6, owing to other skin regions, constricted regions, or background clutter. This may result in a coarse boundary in the finger region with some breaks. If discontinuities occur, the hand's boundary should be smoothed by reducing the energy along it to maintain continuity. Thus, we define a loss function to smooth the boundary of the segmented mask, given by

$$\ell_{\text{smooth}} = \sum_{i,j\in\hat{y}} \sqrt{|(\nabla \hat{y}_{u_{ij}})^2 + (\nabla \hat{y}_{v_{ij}})^2|}, \tag{9}$$

where $u_{i,j}$ and $v_{i,j}$ denote the horizontal and vertical directions of the image coordinates, respectively. ∇ is the gradient operator.

3.4.3. Dice loss with skeletal information

The Dice coefficient estimates the similarity between the predicted mask and ground truth. However, it ignores the hand's shape and continuity. This information is useful because the fingers are slender and often occlude each other, which can cause a segmented region to become disconnected from the main hand segmentation mask. Thus, information about the skeleton of the hand would help the predicted hand regions stay connected despite occlusion, narrow regions, or background clutter. We used the centerline Dice (Shit et al., 2021) with the Dice coefficient to obtain the hand's connectivity and structure. The centerline Dice calculates the skeletons of the predicted mask and the ground truth, denoted by S_P and S_T, respectively. It attends to the intersection of the skeletons with the masks and determines the part

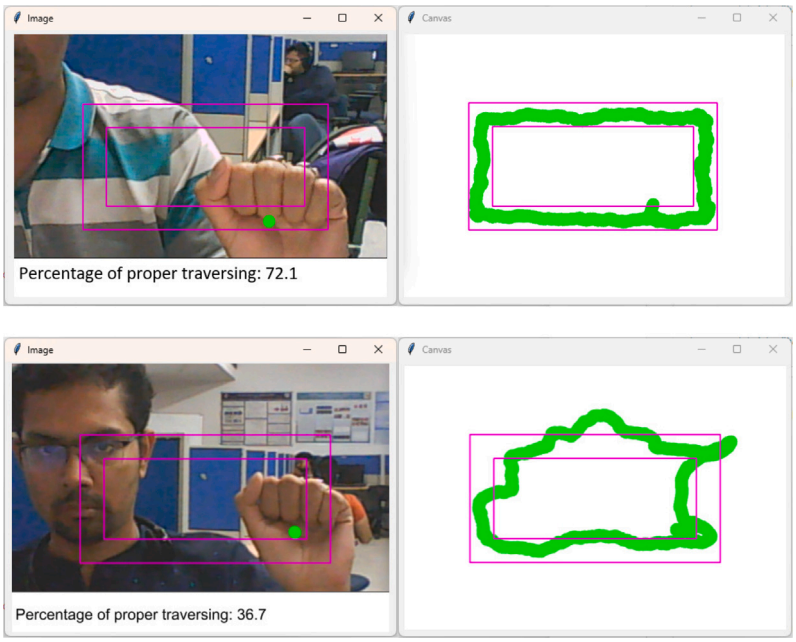


Fig. 8. The interface shows the task of traversing through a predefined boundary. The top row shows proper traversing with a good score, and the bottom row shows improper traversing with a bad score.

of the skeletons that lie in the masks. The skeletal information holds the predicted mask together and iteratively improves it. Thus, the Dice loss with the skeletal information can be defined as

$$\ell_{\text{skdice}} = \kappa_1 \left(1 - \frac{2|\hat{y} \odot y|}{|\hat{y}| + |y| + \epsilon}\right) + \kappa_2 \left(1 - \frac{2(\frac{|S_P \odot y|}{|S_P| + \epsilon} \times \frac{|S_T \odot \hat{y}|}{|S_T| + \epsilon})}{\frac{|S_P \odot y|}{|S_P| + \epsilon}}\right), \tag{10}$$

where κ_1 and κ_2 are constants set to 0.5. Additionally, $\odot$ refers to element-wise multiplication, and $|.|$ refers to the cardinality of the set.

Now, given the three loss functions defined above, we formulate the novel composite loss function given by

$$\ell_{\text{total}} = BFL + \ell_{\text{smooth}} + \ell_{\text{skdice}}. \tag{11}$$

4. Hand rehabilitation tasks

This section describes the interface that uses the hand segmentation results to train users to overcome hand or arm movement difficulties due to injury or other motor complications. The interface asks the users to perform therapeutic tasks involving muscles and nerves, such as flexor and extensor muscles, brachial plexus, radial, ulnar, and median nerves for flexion, extension, abduction, adduction, and other hand movements. The overall layout of the interface is shown in Fig. 7. The user is asked to perform two tasks: traverse a specific path, as shown in Fig. 8, and then grab a virtual object using a gesture and drop it at a specified location on the screen with a different gesture, as shown in Fig. 9. Both tasks are detailed below.

1. Path traversing task: This task trains the user to traverse a defined path while staying within the boundary of the path. The segmented hand mask's centroid (c_x, c_y) is monitored while the hand moves between the two rectangle boundaries. Instances of proper and improper traversing between the boundaries are shown in Fig. 8. The centroid is calculated using (12).

$$c_x = \frac{\sum_x \sum_y x I_s(x,y)}{\sum_x \sum_y I_s(x,y)}, \; c_y = \frac{\sum_x \sum_y y I_s(x,y)}{\sum_x \sum_y I_s(x,y)} \tag{12}$$

where $I_s(x,y)$ is the pixel intensity of the segmented mask I_s at location (x,y). This task enables users to stabilize their movement and exercise their arm muscles. A score is also displayed as a percentage to quantify proper traversing, helping the user to track performance. The score is the number of times the hand's centroid moves within the path divided by the number of times it appears on the screen.

2. Fit and grab task: This task targets hand muscles and nerves affected by a hand injury and improves finger strength. A contour appears at a random location, and the user is asked to trace it. An intersection over union (IoU) score estimates how accurately the user traces the contour. Initially, the user may have difficulty tracing the shape. However, with practice, the user can gradually trace it and improve the IoU score. Once the score is above 60%, the user can grab the object (a rectangle) appearing within the contour, drag it to a specified location, and drop it. To drop the object, the user must close the fist inside a rectangular frame on the screen. The interface depicting this procedure is shown in Fig. 9. Fig. 10 shows an example of a good fit and a bad fit. Only for a good fit does the virtual object appear for grabbing. The contours have different shapes involving different fingers and gesture poses. This ensures the user performs various exercises to strengthen hand and finger operation.

5. Experiments and results

This section describes experiments to evaluate the proposed model, the datasets, and the quantitative and qualitative results. The algorithm is implemented in Python, and the model is trained using an Nvidia Tesla P100 GPU. The training details are shown in Table 1. A uniform image size is chosen for the datasets. The learning rate, batch size, epochs, and d_k are determined using randomized search (Bergstra & Bengio, 2012), and the α and β values are retained from Lin et al. (2017). We initialize a random number generator with a fixed seed to set the initial weights to ensure the results can be reproduced. Also, we use a uniform weight initialization scheme, i.e., Xavier uniform initializer, to maintain near identical variances of its weight gradients across model layers. To quantify model performance, we adopt the

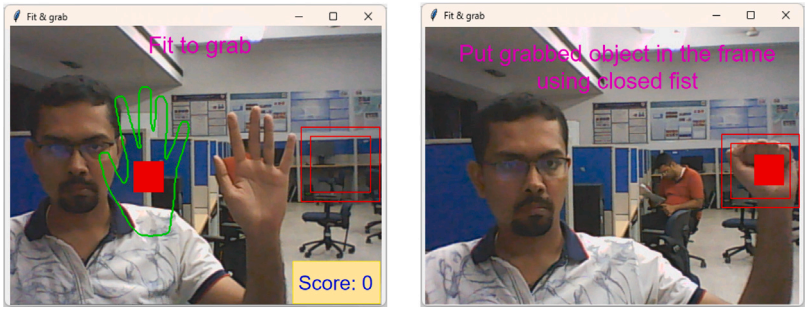


Fig. 9. The interface showing the task of fitting a gesture to grab an object and drop it at a specific location.

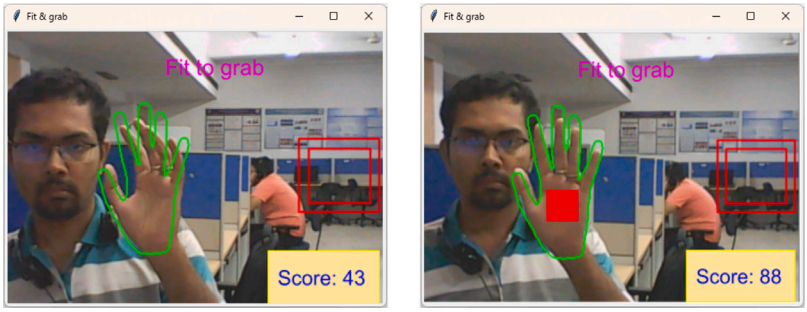


Fig. 10. Left: an instance of bad fit, where the virtual object does not appear. Right: an instance of good fit, which causes the virtual object to appear at the centroid of the hand.

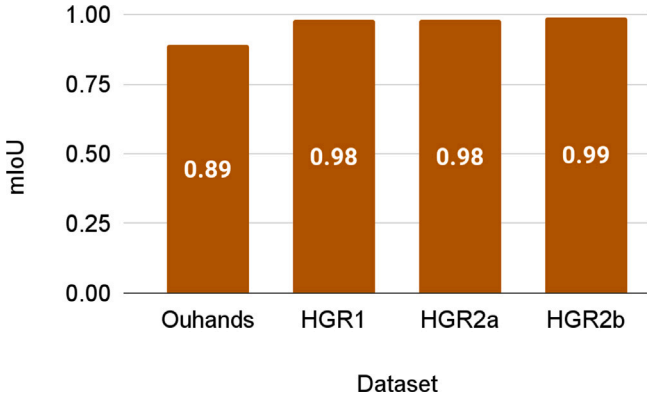


Fig. 11. A plot showing the mIoU score for the four datasets.

F1-score given by

$$F1 - score = \frac{2}{N} \frac{\sum_{i=1}^{N} \frac{M_{ii}^{conf}}{\sum_{j=1}^{N} M_{ji}^{conf}} \times \sum_{i=1}^{N} \frac{M_{ii}^{conf}}{\sum_{j=1}^{N} M_{ij}^{conf}}}{\sum_{i=1}^{N} \frac{M_{ii}^{conf}}{\sum_{j=1}^{N} M_{ji}^{conf}} + \sum_{i=1}^{N} \frac{M_{ii}^{conf}}{\sum_{j=1}^{N} M_{ij}^{conf}}}, \tag{13}$$

where M^{conf} denotes the confusion matrix for the binary class problem (segmentation), and N denotes the number of classes (= 2). Average precision is given by $\frac{1}{N} \sum_i \frac{M_{ii}^{conf}}{\sum_j M_{ji}^{conf}}$, and average recall by $\frac{1}{N} \sum_i \frac{M_{ii}^{conf}}{\sum_j M_{ij}^{conf}}$.

5.1. Datasets

5.1.1. Ouhands

Ouhands (Matilainen et al., 2016) is a hand gesture dataset with around 3000 color images evenly distributed among 10 classes of hand gestures. A total of 23 subjects provided data with varying hand sizes, pose angles, and illumination and with occlusion and background clutter. The dataset comprises annotated hand masks, hand depth images, and each image's annotated hand bounding box coordinates. The images are 480 × 640 pixels.

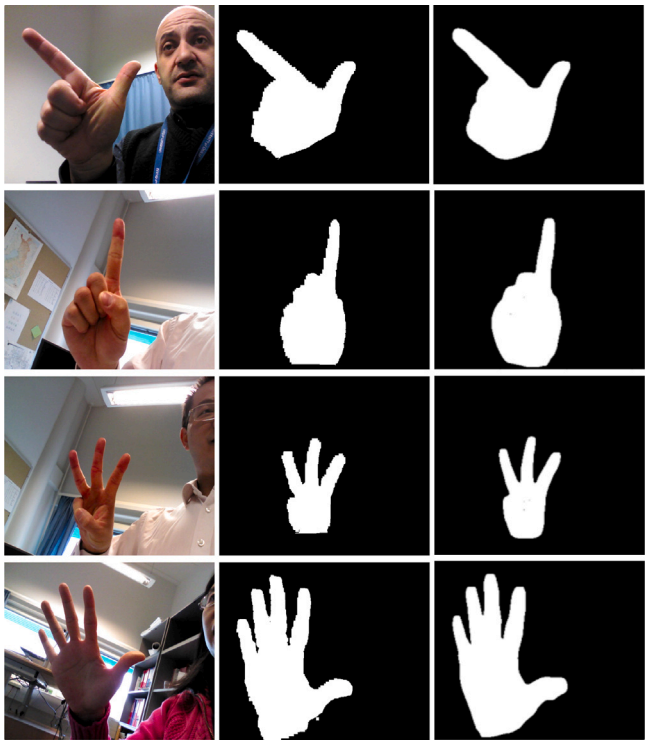
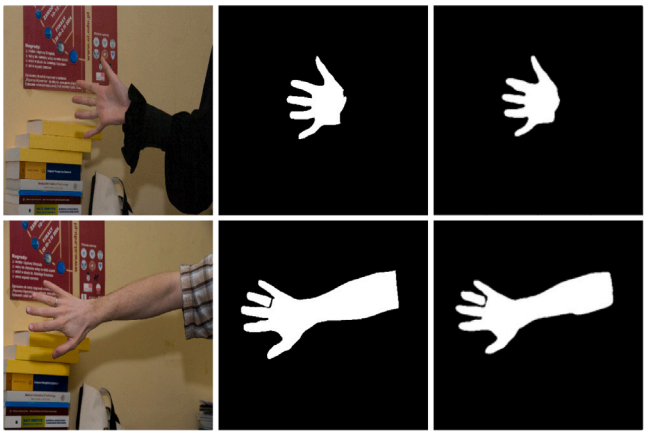
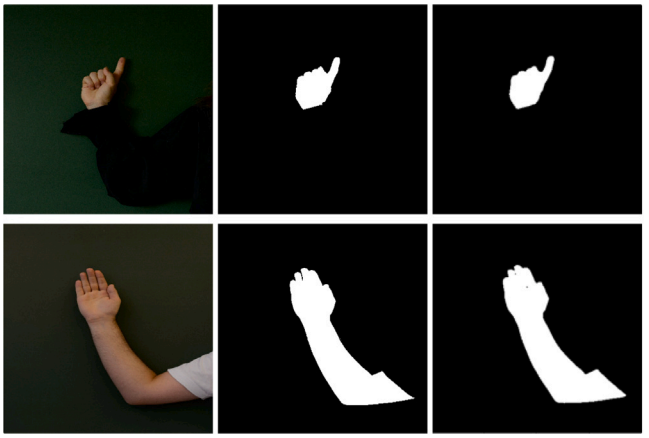


Fig. 12. Segmentation results for the Ouhands dataset. The first column contains the input color image, the second contains the ground truth masks, and the third contains the segmented masks.

(a)



(b)



(c)

Fig. 13. Segmentation results for HGR datasets: (a) HGR1, (b) HGR2a, and (c) HGR2b. The first column contains the input color images, the second contains the ground truth masks, and the third contains the segmented masks.

5.1.2. HGR

The hand gesture recognition dataset (Grzejszczak et al., 2016; Kawulok et al., 2014b; Nalepa & Kawulok, 2014) divides into three parts: HGR1 contains 899 images and 25 classes, HGR2 A contains 85 images and 13 classes, and HGR2B contains 574 images and 32 classes.

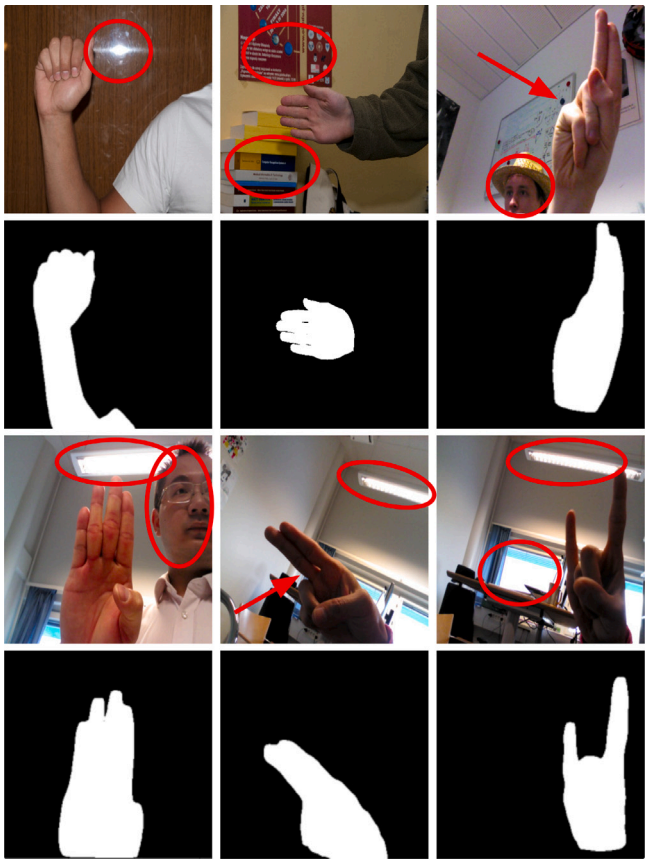


Fig. 14. In these qualitative results, red circles indicate challenging conditions, and arrows indicate specularity and backlighting.

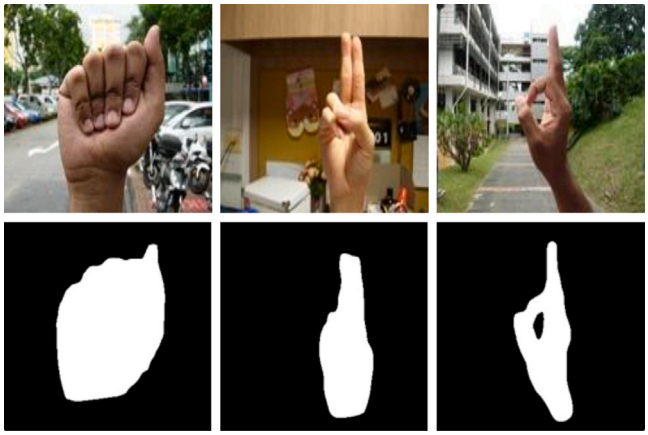


Fig. 15. Segmentation results for the NUS II dataset with a model trained with Ouhands.

Table 1
Training details.

Image shape	$320 \times 320 \times 3$
Optimizer	Adam Optimizer
Learning rate	0.0001
α and β (for BFL)	0.25 and 2 (from Lin et al. (2017))
Batch size	8
Epochs	20
$d_k = d_v$	[16, 64] for encoder,[128, 32] for decoder

Each image has ground truth skin masks and hand keypoint locations. The dataset's resolution, background, and lighting vary by set.

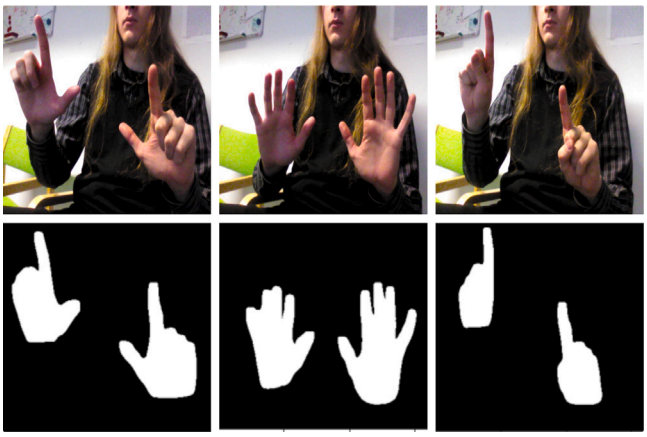


Fig. 16. Segmentation results for images with two hands.

Table 2
Performance comparison for different arrangements of the SFAB and CFAB in the architecture.

Sequence	F1-score (%)	Inference time (ms)
No SFAB or CFAB	95.03	16.27
Only SFAB	95.45	16.15
Only CFAB	95.19	16.35
SFAB → CFAB	97.03	16.54
CFAB → SFAB	96.49	17.05
SFAB ‖ CFAB	96.35	16
SFAB → Conv → CFAB	**97.33**	**15.03**

Table 3
Performance of the model for different numbers of groups created from the SFAB's input feature map.

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

5.2. Placement of self-attention blocks

The SFAB attends to the object's position in the input image, and the CFAB to its class (i.e., foreground or background). The blocks can be arranged sequentially or in parallel to obtain optimal encoder–decoder performance. Thus, we consider six arrangements: No SFAB or CFAB, Only SFAB, Only CFAB, SFAB–CFAB, CFAB–SFAB, SFAB and CFAB in parallel, and SFAB–convolution block–CFAB. Table 2 lists the architecture's performance for these arrangements. Using attention blocks gives better results than using no blocks or only one block. The blocks achieve a better F1-score when arranged in series than in parallel. Moreover, when the SFAB is placed ahead of CFAB, performance improves. Including a convolution block between SFAB and CFAB increases performance by 0.3%. A max-pooling layer was tested in SFAB–Conv–CFAB because it captures class-specific features. Similarly, an upsampling layer was included in the decoder. However, its inclusion had little impact on performance.

Table 3 presents the effect of dividing the SFAB into varying numbers of groups. The F1-score was highest for two groups and then decreased for more groups, while the inference time was lowest for three groups. To maximize accuracy, we divided the incoming feature map into two groups, resulting in an F1-score of 97.33% with an inference time of 15.03 ms. We opted for the best F1-score, giving more weight to segmentation mask accuracy than speed, forfeiting a 0.63 ms decrease in inference time.

Table 4
Performance of the proposed model with different loss functions.

Losses	F1-score (%)
BCE	96.49
BFL	96.93
BFL + ℓ_{smooth}	97.10
BFL + ℓ_{skdice}	97.25
BFL + ℓ_{smooth} + ℓ_{skdice}	**97.33**

Table 5
Comparison of different baseline models with the proposed model.

Methods	F1-score (%)	mIoU (%)	Inference time (ms)
U-Net (Ronneberger et al., 2015)	96.7	81.2	25
Attention U-Net (Oktay et al., 2018)	96.8	85.5	29
RA-UNet (Jin et al., 2020)	96.9	84.9	32
FCN-8s (Long et al., 2015)	95.5	80.3	63
CBAM (Woo et al., 2018)	96.3	82.9	38
DANet (Fu et al., 2019)	96.4	83.6	35
Ours	**97.3**	**89**	**15**

5.3. Performance of the loss components

This subsection explains how the loss functions contributed to achieving a state-of-the-art result. Because BFL performed better than BCE, the contributions of ℓ_{smooth} and ℓ_{skdice} were tested for BFL. Further experimentation revealed that, although ℓ_{smooth} and ℓ_{skdice} contributed to a good F1-score, the combination of BFL, ℓ_{smooth}, and ℓ_{skdice} resulted in even better performance. Table 4 lists the model's performance for different loss functions.

5.4. Comparison with baseline models

The proposed model's performance was validated by comparing it with a few baseline architectures. The performance of encoder–decoder models with attention, such as Attention U-Net (Oktay et al., 2018) and RA-UNet (Jin et al., 2020) surpassed the performance of simple encoder–decoder models, such as U-Net (Ronneberger et al., 2015) and FCN (Long et al., 2015). Therefore, we included more attention-based models for comparison, such as CBAM (Woo et al., 2018) and DANet (Fu et al., 2019). We chose these six deep neural networks as baselines because their encoder–decoder architecture with attention is similar to ours. A novel network should perform better than the baselines to show its contribution. The results in Table 5 indicate that the proposed network performed better than the baselines, considerably improving inference time. Inference time was 10 ms faster than U-Net, the second-fastest architecture, because of our model's efficient spatial and channel attention. The F1-score was 0.6% higher than RA-UNet, which had the second-highest F1-score. Also, the mIoU value was much higher than for other models, 3.5% higher than the second highest.

Moreover, we conducted a Friedman tests to determine whether differences in model performance were statistically significant. Table 6 compares the models for the Ouhands, HGR, and NUS datasets, obtaining a p-value of 0.000335. Since the p-value is less than the alpha value (= 0.05), we can reject the null hypothesis and conclude that the difference in model performance is significant.

5.5. State-of-the-art comparison

We also compared the proposed model with state-of-the-art hand segmentation approaches. Tables 6 and 7 list the performance of the state-of-the-art methods for the Ouhands and HGR datasets, respectively. For the Ouhands dataset, the proposed model and DeepLabv3 (Chen et al., 2017) were tied for the highest F1-score at 97.3%. However, the mIoU value of 89% is the best among other competing models. At 15 ms, the proposed model outperformed all others for inference

Table 6
Comparison of the proposed method and state-of-the-art methods for the Ouhands dataset.

Methods	F1-score (%)	mIoU (%)	Inferencetime (ms)	#Parameters
FCN-8s (Long et al., 2015)	95.5	80.3	63	134 M
PSPNet (Zhao et al., 2017)	97.0	80.2	50	79.44 M
DeepLabv3 (Chen et al., 2017)	**97.3**	87.3	43	75.30 M
HGR-Net (Dadashzadeh et al., 2019)	96.3	82.6	21	**0.28 M**
CBAM (Woo et al., 2018)	96.3	82.9	38	7.97 M
DANet (Fu et al., 2019)	96.4	83.6	35	7.56 M
U-Net (Ronneberger et al., 2015)	96.7	81.2	25	7.86 M
Segment Anything (Kirillov et al., 2023)	79.44	79.77	38	641.09 M
Ours	**97.3**	**89.0**	**15**	1.02 M

Table 7
Comparison of the proposed method and state-of-the-art methods for the HGR dataset.

Methods	F1-score (%)	mIoU (%)	Inferencetime (ms)	#Parameters
Hettiarachchi and Peters (Hettiarachchi & Peters, 2016)	96.9	–	–	–
Kawulok et al. (2014a)	95.6	–	–	–
Kawulok (2013)	90.8	–	–	–
FCN-8s (Long et al., 2015)	97.7	91.2	63	134 M
PSPNet (Zhao et al., 2017)	98.7	94.0	50	79.44 M
DeepLabv3 (Chen et al., 2017)	98.8	97.2	43	75.30 M
HGR-Net (Dadashzadeh et al., 2019)	98.2	95.8	21	**0.28 M**
OR-Skip_net (Arsalan et al., 2020)	96.9	94.3	38	9.72 M
Lumini et al. (Lumini & Nanni, 2020)	96.7	–	–	–
Segment Anything (Kirillov et al., 2023)	92.8	94.7	39	641.09 M
Ours	**99.3**	**98.3**	**10**	1.02 M

time. The proposed model had the second-fewest parameters after HGR-Net (Dadashzadeh et al., 2019).

For the HGR dataset, the proposed model performed phenomenally, attaining an F1-score of 99.3% and mIoU of 98.3% in 10 ms, surpassing state-of-the-art methods.

Fig. 11 shows the mIoU scores for the datasets used. mIoU is a useful metric to quantify the accuracy of the segmented mask, and is given by

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{M_{ii}^{conf}}{\sum_{j=1}^{N}M_{ij}^{conf}+\sum_{j=1}^{N}M_{ji}^{conf}-M_{ii}^{conf}}. \quad (14)$$

The mIoU scores for the Ouhands, HGR1, HGR2a, and HGR2b datasets were 0.89, 0.98, 0.98, and 0.99, respectively.

5.6. Qualitative assessment

Figs. 12–17 show the qualitative results of the proposed method. Fig. 12 displays the segmentation results for the Ouhands dataset. The results show the hand's shape (column 3). It has a smooth boundary without any gaps. Similarly, Fig. 13 shows the predicted segmentation masks for the HGR datasets (column 3). The proposed model performed exceptionally well for images with different hand poses and orientations, the presence of a human face and background clutter, variations in illumination, and specularity. This is shown in Fig. 14, which highlights challenging conditions with an arrow or circle. Owing to the attention mechanism, the model does not deviate from the region of interest. Normalization and using geometrically altered training samples make the model robust against changes in light, orientation, and scale.

Moreover, we tested the model trained with the Ouhands dataset with the NUS II dataset (Pisharady et al., 2013). NUS II is a 10-class hand gesture recognition dataset that lacks segmentation masks.

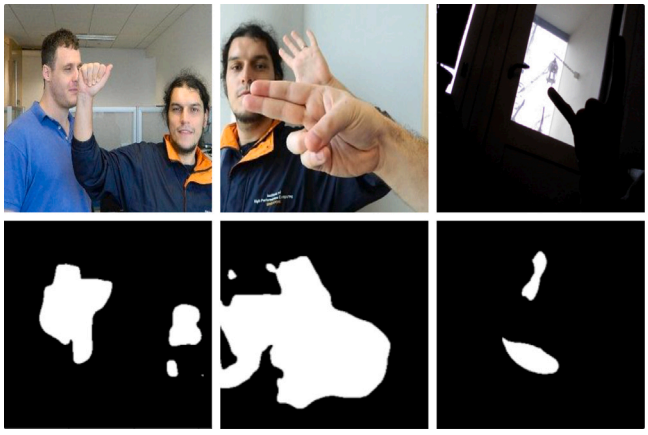


Fig. 17. Some failed segmentation cases.

However, the segmentation results in Fig. 15 were encouraging. Despite background clutter, the model performed well. The model was also tested on supplementary data samples from Ouhands containing two hands in an image. Fig. 16 shows that the segmentation masks are precisely detected even for two hands.

Thus, the quantitative and qualitative results highlight the proposed method's ability to perform hand segmentation accurately and efficiently. However, the model failed under certain conditions, as shown in Fig. 17. For example, it failed when the hand and face were the same color and overlapping (column 1), when the hands and face lacked distinct hand regions (column 2), and when darkness made the hand almost unrecognizable (column 3). These issues must be addressed in future work.

Table 8
Performance tracking for five weeks of five users who received training for the two hand rehabilitation tasks.

	Traversing score (%)					IoU score (approx. %) in time (s)				
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 1	Week 2	Week 3	Week 4	Week 5
User 1	20.1	29.5	32.6	45.7	57.9	60 in 5	67 in 5	75 in 3.5	82 in 1.5	87 in 1.5
User 2	35	45.3	50.1	59.2	68.2	61 in 5	70 in 4	81 in 4	83 in 1	83 in 0.5
User 3	33.7	45	54.9	62.1	68.8	70 in 4.5	75 in 3	83 in 3	85 in 3	91 in 1
User 4	27.8	35.7	42.7	51.7	58.8	65 in 6	73 in 3	82 in 1.5	88 in 1	88 in 1
User 5	35.6	38.6	47.5	52.1	57.3	64 in 5.5	70 in 3	86 in 1.5	89 in 1	92 in 1
Mean	30.44	38.82	45.56	54.16	62.20	64 in 5.2	71 in 3.6	81.4 in 2.7	85.4 in 1.5	88.2 in 1
Standard Deviation	5.86	5.95	7.59	5.83	5.17	3.52 in 0.51	2.75 in 0.8	3.61 in 1.03	2.73 in 0.77	3.19 in 0.32

Table 9
Performance tracking for five weeks of five users who received no training for the two hand rehabilitation tasks.

	Traversing score (%)					IoU score (approx. %) in time (s)				
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 1	Week 2	Week 3	Week 4	Week 5
User 1	23.2	24.5	25.7	29.6	33.5	58 in 7.5	59 in 8	60 in 5	63 in 5	65 in 4
User 2	30.9	31.3	32.8	35.5	40.1	59 in 8	59 in 7	63 in 6	66 in 6.5	66 in 4
User 3	32.2	32.4	32.9	36.7	41.8	62 in 8	62 in 6	65 in 5	66 in 5	69 in 3
User 4	25.6	26.5	27.5	30.6	37.2	62 in 8.5	62 in 6.5	65 in 5	65 in 3	68 in 3.5
User 5	31.2	31.9	33.1	36.6	40.1	60 in 7	60 in 6	60 in 4.5	63 in 2	65 in 2
Mean	28.62	29.32	30.4	33.8	38.54	60.2 in 7.8	60.4 in 6.7	62.6 in 5.1	64.6 in 4.3	66.6 in 3.3
Standard Deviation	3.55	3.20	3.15	3.06	2.92	1.35 in 0.51	2.24 in 0.75	1.35 in 0.49	1.62 in 1.60	1.59 in 0.75

5.7. Rehabilitation tasks' assessment

Ten patients aged 25 to 35 with hand and arm mobility issues, were recruited to interact with the interface. The patients were randomly assigned to two groups, with five in each group. Only one group received training on the tasks. The users gave informed consent to participate in the study, which IIT Guwahati's ethics committee approved. The users' performance was tracked for five weeks for both groups' traversing and grabbing tasks, as shown in Tables 8 and 9, respectively. The group that received training was trained with the interface's tasks three to four days a week and was asked to test themselves on one of the remaining days. However, the group without training was asked to perform the tasks on any day of the week. The users who received training improved gradually by performing the rehabilitation tasks, and most regained normal hand function. Their traversing and IoU scores in the first week were low, and task completion was slow. However, by week five, they had significantly improved their scores and time to completion. The users from the other group who did not receive training showed minimal improvement in performing their tasks. The performance of the two groups was compared. The p-value for traversing scores was $p\text{-value}_{\text{traversing scores}} = 0.045$ and the p-value for IoU scores was $p\text{-value}_{\text{IoU scores}} = 0.012$ indicating that consistent training with the rehabilitation interface resulted in a statistically significant improvement.

6. Conclusion

We devised an interface for hand rehabilitation that combines rehabilitation tasks with hand segmentation. For its robust operation, the segmentation should be efficient and accurate. Therefore, we have made the following contributions:

- We designed a deep neural network with an attention mechanism that outputs accurate hand segmentation masks.
- To address the memory requirements of the self-attention mechanism, we proposed two efficient attention blocks, the SFAB and CFAB, to calculate attention in two dimensions–spatial and channel–that are combined with the encoder–decoder architecture.
- We defined a composite loss function to optimize the model.
- We developed a hand rehabilitation interface comprising two rehabilitation tasks that helps patients regain normal hand or arm movement.

The qualitative results demonstrate the model's ability to obtain precise hand segmentation masks in challenging environments without prior training. The model also generated masks when two hands were present. The quantitative results support the model's segmentation accuracy and efficiency. It achieves an F1-score of 97.3% and 99.3% for Ouhands and HGR datasets, respectively. It also performs better than state-of-the-art methods with higher F1-scores and faster inference times. The rehabilitation tasks showed their effectiveness by gradually helping the users regain normal hand mobility.

Future Work: This model needs improvement to segment overlapping hands and faces or in extremely low illumination. Therefore, we would work towards localizing the hand in such extreme cases. Hand segmentation capturing the temporal information of a video is another aspect we would consider for future development of the work. Moreover, we would incorporate classification with segmentation, increasing the scope of including more therapeutic activities for hand rehabilitation. However, real-time implementation would get affected in such cases, which we would further investigate.

CRediT authorship contribution statement

H Pallab Jyoti Dutta: Conceptualization, Methodology, Software, Writing – original draft, Visualization. **M.K. Bhuyan:** Validation, Formal analysis, Supervision, Funding acquisition. **Debanga Raj Neog:** Conceptualization, Writing – original draft. **Karl Fredric MacDorman:** Validation, Writing – review & editing. **Rabul Hussain Laskar:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used

Acknowledgment

We acknowledge the Department of Biotechnology, Government of India for the financial support for the Project BT/COE/34/SP28408/2018.

References

Adaloglou, N., Chatzis, T., Papastratis, I., Stergioulas, A., Papadopoulos, G. T., Zacharopoulou, V., Xydopoulos, G. J., Atzakas, K., Papazachariou, D., & Daras, P. (2022). A comprehensive study on deep learning-based methods for sign language recognition. *IEEE Transactions on Multimedia*, *24*, 1750–1762. http://dx.doi.org/10.1109/TMM.2021.3070438.

Almeida, A., Vicente, P., & Bernardino, A. (2021). Where is my hand? Deep hand segmentation for visual self-recognition in humanoid robots. *Robotics and Autonomous Systems*, *145*, Article 103857. http://dx.doi.org/10.1016/j.robot.2021.103857, URL: https://www.sciencedirect.com/science/article/pii/S0921889021001421.

Arsalan, M., Kim, D. S., Owais, M., & Park, K. R. (2020). OR-Skip-Net: Outer residual skip network for skin segmentation in non-ideal situations. *Expert Systems with Applications*, *141*, Article 112922. http://dx.doi.org/10.1016/j.eswa.2019.112922.

Baheti, B., Innani, S., Gajre, S., & Talbar, S. (2020). Eff-UNet: A novel architecture for semantic segmentation in unstructured environment. In *2020 IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 1473–1481). http://dx.doi.org/10.1109/CVPRW50498.2020.00187.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(null), 281–305.

Cai, M., Lu, F., & Sato, Y. (2020). Generalizing hand segmentation in egocentric videos with uncertainty-guided model adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Chakraborty, B., & Bhuyan, M. (2020). Image specific discriminative feature extraction for skin segmentation. *Multimedia Tools and Applications*, *79*, 18981–19004. http://dx.doi.org/10.1007/s11042-020-08762-4.

Chakraborty, B. K., Sarma, D., Bhuyan, M., & MacDorman, K. F. (2018). Review of constraints on vision-based gesture recognition for human–computer interaction. *IET Computer Vision*, *12*(1), 3–15. http://dx.doi.org/10.1049/iet-cvi.2017.0052.

Chen, L., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. CoRR abs/1706.05587, arXiv:1706.05587.

Dadashzadeh, A., Targhi, A., Tahmasbi, M., & Mirmehdi, M. (2019). HGR-Net: A fusion network for hand gesture segmentation and recognition. *IET Computer Vision*, *13*(8), 700–707. http://dx.doi.org/10.1049/iet-cvi.2018.5796.

Dutta, H. P. J., Sarma, D., Bhuyan, M., & Laskar, R. H. (2020). Semantic segmentation-based hand gesture recognition using deep neural networks. In *2020 national conference on communications* (pp. 1–6). http://dx.doi.org/10.1109/NCC48643.2020.9055990.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. (CVPR).

Grzejszczak, T., Kawulok, M., & Galuszka, A. (2016). Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, *75*(23), 16363–16387. http://dx.doi.org/10.1007/s11042-015-2934-5.

Hettiarachchi, R., & Peters, J. (2016). Multi-manifold-based skin classifier on feature space voronoïregions for skin segmentation. *Journal of Visual Communication and Image Representation*, *41*(C), 123–139. http://dx.doi.org/10.1016/j.jvcir.2016.09.011.

Jin, Q., Meng, Z., Sun, C., Cui, H., & Su, R. (2020). RA-UNet: A hybrid deep attention-aware network to extract liver and tumor in CT scans. *Frontiers in Bioengineering and Biotechnology*, *8*, http://dx.doi.org/10.3389/fbioe.2020.605132.

Ju, Z., Ji, X., Li, J., & Liu, H. (2017). An integrative framework of human hand gesture segmentation for human–robot interaction. *IEEE Systems Journal*, *11*(3), 1326–1336. http://dx.doi.org/10.1109/JSYST.2015.2468231.

Kawulok, M. (2013). Fast propagation-based skin regions segmentation in color images. In *2013 10th IEEE international conference and workshops on automatic face and gesture recognition*.

Kawulok, M., Kawulok, J., & Nalepa, J. (2014). Spatial-based skin detection using discriminative skin-presence features. *Pattern Recognition Letters*, *41*, 3–13. http://dx.doi.org/10.1016/j.patrec.2013.08.028.

Kawulok, M., Kawulok, J., Nalepa, J., & Smolka, B. (2014). Self-adaptive algorithm for segmenting skin regions. *EURASIP Journal on Advances in Signal Processing*, *2014*(170), 1–22. http://dx.doi.org/10.1186/1687-6180-2014-170.

Khan, A. U., & Borji, A. (2018). Analysis of hand segmentation in the wild. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 4710–4719).

Khan, R., Hanbury, A., Stöttinger, J., & Bais, A. (2012). Color based skin classification. *Pattern Recognition Letters*, *33*(2), 157–163. http://dx.doi.org/10.1016/j.patrec.2011.09.032.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). Segment anything. arXiv:2304.02643.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *2017 IEEE international conference on computer vision* (pp. 2999–3007). http://dx.doi.org/10.1109/ICCV.2017.324.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 IEEE conference on computer vision and pattern recognition* (pp. 3431–3440). http://dx.doi.org/10.1109/CVPR.2015.7298965.

Lumini, A., & Nanni, L. (2020). Fair comparison of skin detection approaches on publicly available datasets. *Expert Systems with Applications*, *160*, Article 113677. http://dx.doi.org/10.1016/j.eswa.2020.113677.

Luo, Z., Lim, C. K., Yang, W., Tee, K. Y., Li, K., Gu, C., Nguen, K. D., Chen, I.-M., & Yeo, S. H. (2010). An interactive therapy system for arm and hand rehabilitation. In *2010 IEEE conference on robotics, automation and mechatronics* (pp. 9–14). http://dx.doi.org/10.1109/RAMECH.2010.5513222.

Matilainen, M., Sangi, P., Holappa, J., & Silvén, O. (2016). OUHANDS database for hand detection and pose recognition. In *2016 sixth international conference on image processing theory, tools and applications* (pp. 1–5). http://dx.doi.org/10.1109/IPTA.2016.7821025.

Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *37*(3), 311–324. http://dx.doi.org/10.1109/TSMCC.2007.893280.

Nalepa, J., & Kawulok, M. (2014). Fast and accurate hand shape classification. In S. Kozielski, D. Mrozek, P. Kasprowski, B. Malysiak-Mrozek, & D. Kostrzewa (Eds.), *Communications in computer and information science*: *vol. 424*, *Beyond databases, architectures, and structures* (pp. 364–373). Springer, http://dx.doi.org/10.1007/978-3-319-06932-6_35.

Ohkawa, T., Yagi, T., Hashimoto, A., Ushiku, Y., & Sato, Y. (2021). Foreground-aware stylization and consensus pseudo-labeling for domain adaptation of First-Person hand segmentation. *IEEE Access*, *9*, 94644–94655. http://dx.doi.org/10.1109/ACCESS.2021.3094052.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). Attention U-Net: Learning where to look for the pancreas. arXiv:1804.03999.

Pisharady, P. K., Vadakkepat, P., & Loh, A. P. (2013). Attention based detection and recognition of hand postures against complex backgrounds. *International Journal of Computer Vision*, *101*(3), 403–419. http://dx.doi.org/10.1007/s11263-012-0560-5.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention* (pp. 234–241). Cham: Springer International Publishing.

Shit, S., Paetzold, J. C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J. P., Bauer, U., & Menze, B. H. (2021). clDice-A novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16560–16569).

Sun, X., Sun, X., Chen, C., Wang, X., Dong, J., Zhou, H., & Chen, S. (2021). Gaussian dynamic convolution for efficient single-image segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 1. http://dx.doi.org/10.1109/TCSVT.2021.3096814.

Tsai, T.-H., & Huang, S.-A. (2022). Refined U-net: A new semantic technique on hand segmentation. *Neurocomputing*, *495*, 1–10. http://dx.doi.org/10.1016/j.neucom.2022.04.079.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems, vol. 30*. Curran Associates, Inc..

Wang, Y., Peng, C., & Liu, Y. (2019). Mask-pose cascaded CNN for 2D hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(11), 3258–3268. http://dx.doi.org/10.1109/TCSVT.2018.2879980.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. In *Proceedings of the European conference on computer vision*.

Wu, H., Zhang, J., Huang, K., Liang, K., & Yizhou, Y. (2019). FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation. arXiv preprint arXiv:1903.11816.

Yang, F., & Wu, Y. (2019). A Soft Proposal Segmentation Network (SPS-Net) for hand segmentation on depth videos. *IEEE Access*, *7*, 29655–29661. http://dx.doi.org/10.1109/ACCESS.2019.2900991.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *2017 IEEE conference on computer vision and pattern recognition* (pp. 6230–6239). http://dx.doi.org/10.1109/CVPR.2017.660.

Zhuoran, S., Mingyuan, Z., Haiyu, Z., Shuai, Y., & Hongsheng, L. (2021). Efficient attention: Attention with linear complexities. In *2021 IEEE winter conference on applications of computer vision* (pp. 3530–3538). http://dx.doi.org/10.1109/WACV48630.2021.00357.