IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 3, June, 2022, pp. 1039-1050
Journal homepage: http://iieta.org/journals/ts

Hand Gesture Recognizing Model Using Optimized Capsule Neural Network

Suni S S[1*], K Gopakumar[2]

[1] LBS Centre for Science & Technology, University of Kerala, Kerala 695033, India
[2] APJ Abdul Kalam Technological University, Thiruvananthapuram, Kerala 695016, India

Corresponding Author Email: suni.ss@gmail.com

https://doi.org/10.18280/ts.390331

Received: 23 April 2022
Accepted: 13 May 2022

Keywords:
hand gestures, human-computer interface (HCI), deep learning, SoftMax layer, capsule neural network (CapsNet)

ABSTRACT

Hand gestures are a sort of nonverbal communication that may be utilized for many diverse purposes, including deaf-mute interaction, robotic manipulation, human-computer interface (HCI), residential management, and healthcare usage. Moreover, most current research uses the artificial intelligence approach effectively to extract dense features from hand gestures. Since most of them used neural network models, the performance of the models influences the modification of the hyperparameter to enhance recognition accuracy. Therefore, our research proposed a capsule neural network, in which the internal computations on the inputs are better encapsulated by transforming the findings into a tiny vector of information outputs. Moreover, to increase the accuracy of recognizing hand gestures, the neural network has been optimized by inserting additional SoftMax layers before the output layer of the CapsNet. Subsequently, the findings of the tests were assessed and then compared. This developed approach has been beneficial across all tests when contrasted against state-of-the-art systems.

1. INTRODUCTION

Human-computer interaction had already evolved immensely, and now the field has always been progressing, with those fresh ideas and methodologies getting created. Gestures were spontaneous expressions from the human figure utilized effectively to interact with those around [1, 2]. This hand gesture seems to be the most extensively employed interaction among the many motions. It must be viewed as a distinct hand movement at a given moment. Signals were employed in almost two-thirds of overall discussions [3]. Hand gesture recognition enables one may create innovative, extra naturalistic strategies for human-machine conversations. The human component of the interaction is crucial to gesture recognition models. Hand gesture acknowledgement offers an incredible ability to revolutionize human-computer interaction (HCI). It also facilitates communication, particularly between the deaf population and the broader public. Visual experience now plays an important role in HCIs. Human behaviour can be recognized as an input for processing by interactive and appropriately gesture-classifiable computer software. The employment of sign language as a gesture-based, rather than voice-based, mode of communication with other media makes HCI extremely potential. Hand gesture detection is among the greatest broad fields wherein machine vision and intelligent systems have also enhanced interaction among deaf people and enabled gesture-depended signaling models [4, 5].

Hand gestures were categorized as static as well as dynamic. The authors [6] indicate that static gestures essentially use a single posture retained for a fixed period to convey the desired information. An example of a static gesture is the user acknowledging in response to an application confirmation query may be coupled with the "thumbs up" posture retained for 1 second. A dynamic gesture is characterized as a continuous or discrete function of time. For recognizing the significance of the user's gesture command (for example, the "drag & drop" command), dynamic generalized gestures are gestures in which both the motion trajectory and posture are equally essential. The primary distinction between posture and gesture seems to be that posture typically emphasizes hand contour, whereas gesture emphasizes hand motion. Portable gloves-depended sensor approaches and Camera perception-dependent methodologies seem to be the two massive technologies for hand gestures investigation [6, 7].

Wearable sensors inserted firmly on hand-wearing gloves have been utilized to identify hand motions. Hand motions or finger bending cause these sensors to detect a physical reaction. The same information is again processed on a computer coupled with gloves. Besides, integrating a sensor into either a microcontroller or a glove-depended sensing technology might have been portable [8-13]. Although the approaches described above have shown positive results, they have certain drawbacks that make them inappropriate for the elderly, who may feel discomfort and disorientation due to cable connection issues. These disadvantages can be solved by using vision-based approaches. Identifying hand gestures makes it possible to establish novel and increasingly naturalistic approaches involving human-machine contact.

Approaches for gesture recognition depend substantially just on the individual element aspect of the interactions. Establishing an adequate human-computer interaction enabling sign language recognition is a vital domain of work that has already resulted in professional life aspirations, including for programmers and consumers. To that end, research into the design of computer assistive technologies is gaining attraction and becoming increasingly crucial. Also, the evolution of gesture detection technologies was particularly indispensable towards the progress of computers and the social

interface, including the employment of hand gestures, was becoming much more prevalent in many fields. Hand gesture identification encompasses subdomains including certain sign language acknowledgement [14-16], identification of particularly unique signal language utilized throughout sports [17], complete human motion diagnosis [18], pose as well as body position identification [19, 20], physical activity actively supervising [21], and sometimes even attempting to regulate smart residence living applications only with hand gestures [22]. Hand size diversity, skin texture and colouring, lighting, viewpoint discrepancy, resemblance in diverse motions, and even the crucial ecological context pose difficult obstacles toward vision-dependent hand gesture detection. Moreover, recognizing dynamic hand gestures in images can be difficult due to the different and diverse circumstances depicted in the images.

Hand gesture tracking has already gotten quite a lot of enthusiasm in communication and machine learning disciplines. The basic idea in the machine learning area for HCI is to identify and detect human gestures accurately. Hand gesture identification's definite purpose should be to distinguish and acknowledge gestures. Hand gesture detection is a path that incorporates numerous notions, including methods throughout diverse fields, like image processing and neural networks, which effectively interpret how well a hand moves. Hand-crafted feature extraction approaches using classifiers were a lot more popular and were commonly used strategies for hand gesture detection before the emergence of deep learning [23-25].

A Support Vector Machine (SVM) was being used to categorize Local Binary Patterns (LBP) characters [26]; Histograms of such oriented gradations (HOG) as well as LBP are being used in the study of Jadooki et al. [27]. Moreover, to extract the attribute features, Nakjai and Katanyukul [28] used an Artificial Neural Network (ANN) with discrete cosine transformation (DCT), as well as CNN iterations for sign language acknowledgement were presented in the study of Perimal et al. [29]. Chen et al. [30] suggests a multiscale attention fusion network for semantic segmentation (MAF-DeepLab), which emphasizes key aspects and effectively combines multi-scale information, to enhance prediction performance. In contrast, several aforementioned neural network variations had a decent recognition rate. However, it employs a single scalar output to summarize a pool of repeated local characteristics, and it does not do internal calculations on the input.

The target of the whole work should be to design robust deep learning-dependent hand gesture detection technology that would be leveraged to integrate with enhanced reality apps. Consequently, this research proposed a capsule neural network that identifies hand movements effectively and produces better results than the existing studies.

- Our research proposed a novel capsule neural network to avoid the generalization problem in neural networks. The capsules do substantial internal calculations on the inputs to better encapsulate the findings by converting the outputs into a compact vector of information outputs.
- Furthermore, the CapsNet structure is modified by introducing additional SoftMax layers before the output layer to optimize the neural network process, which improves the accuracy of recognizing hand gestures.

The accompanying structure of the whole research work: An assessment of neural nets have been addressed in the second section. The third section describes the concepts of such a capsule neural net. This proposed method system and its architecture are detailed in the fourth section. The final section covered the execution as well as comparing outcomes. Also, the final section concludes the proposed work.

2. LITERATURE SURVEY

As human-computer interaction expands, several solutions are leveraging combined machine learning and deep learning approaches targeted at detecting a gesture intonated by a person's hand. Furthermore, several publications are being reviewed to understand the functioning of such a hand gesture detection approach.

Perimal et al. [29] developed a hand gesture identification algorithm for fourteen hand gestures based on finger counting. The algorithm uses the maximum distance between the centroid of the fingers to count fingers and detect the hand gestures. To further understand the algorithm's performance, the developed method by the author [29] was tested using a variety of prospect and dynamic parameters. However, the developed hand gesture identification algorithm is less accurate.

Employing static RGB-D images, Li et al. [31] presented an effective deep attentiveness network enabling joint hand gesture tracking and detection. The model automatically locates the hand without geographical annotations and performs exceptionally well in gesture classification. This strategy, however, will not work for dynamic motions.

Alani et al. [32] proposed an Adapted Deep Convolutional Neural Network (ADCNN) Infrastructure adequate for the categorization of fixed hand motion picture data with distinctions throughout illumination, noise, magnitude, spinning, as well as transcription, where information enhancement was used to generate morphing pictures from actual pictures, and the enhanced images were then used to boost the learning algorithm. Moreover, ADCNN will be enhanced and evaluated with other datasets in the future and real-time hand gesture classification tasks.

Nunez et al. [33] formulated a Deep Learning (DL)-dependent strategy for sequential 3D shape pattern identification predicated on the same incorporation of Convolutional Neural Network (CNN) as well as Long Short Term Memory (LSTM) recurrent networks and also the data enhancement methodology that either strengthens the effectiveness of a certain system or inhibits overfitting to the greatest extent possible. However, for small datasets, the proposed data augmentation stage has a higher impact on the learning process, resulting in the greatest performance; the training time will increase for large datasets.

Li et al. [34] developed a Deep Convolution Network-dependent motion identification approach. The properties of CNN are employed to prevent feature extraction and diminish the count of variables that should be learned, eventually attaining the purpose of further unsupervised learning. This same error back-propagation technique was given into CNN architecture, which adjusts the threshold and parameters of the same neural net to lessen system errors. The suggested SVM is being utilized to boost the overall model's reliability and endurance by optimizing its categorization functionality of a CNN model. In the future, use filter size to assess current content in most prior and future data. However, in the long dependence, CNN is not as good as LSTM because size is a constant value.

Cheng et al. [35] concentrated on static gesture detection, using the Kinect sensor to gather colour and depth gesture samples, then analyzed. On this premise, a CNN-RBM collaborative network for gesture recognition is proposed. It primarily leverages numerous RBMs' enclosed systems for unsupervised retrieval of features, ultimately merging with CNN's supervised attribute retrieval. Eventually, those two characteristics were linked to categorizing hand gestures. Even though the joint network and other centralized networks perform poorly on the complicated sample because RBM necessitates appropriate data distribution. As a result, future research will focus on improving the combined network's accuracy in a complicated configuration.

Skaria et al. [36] used a tiny radar sensor to gather Doppler signers of 14 distinct hands and maybe even train a deep CNN to distinguish such seized motions, in which the rhythmic signals from the two acquiring antennas of either long ongoing-wave radar equipment were able to create the in-phase, as well as quadrature elements of the rhythmic signals, were being used to develop feature arrays. As a result, it appears that various users can use this architecture. Also, in future, the authors plan to increase the overall accuracy of the hand gestures.

Wu [37] developed a novel identification approach that relies on the dual-channel CNN (DC-CNN) model to boost detection rates. The CNN's two distinct input channels are mostly hand motion pictures and hand edge pictures, while features fusion was done just at the entire connectivity tier. A SoftMax model categorized those gestures. There is still a lot of scope for DC-CNN research and development, especially in the three phases described below. (1) Incorporate more hierarchy and scale elements to enhance the model's adaptability to complicated backgrounds. (2) There is still a lot of untapped potential in the rate of dynamic gesture recognition, and the model can be used in this field. (3) For training, the convolution neural network model for gesture identification necessitates many image data with labels. The training could be done using unsupervised or semi-supervised learning to eliminate the model's reliance on a huge quantity of tag data.

Qi et al. [38] developed an identification technique integrating principal element analytics and a General Recursive Neural Network (GRNN). Such an approach seems effective for lessening signal dimensionality, enhancing overall accuracy, or even effectiveness for real-time diagnosis, including retrieving key data essential during individual physical movement to discover discrete movement motions, which are being used to derive face EMG attributes. Before actually creating a GRNN neural net, PCA has been used to diminish feature granularity via omitting extraneous data. This GRNN paradigm should assist in the detection of the most precise form of hand movements, contributing to clinical medicines, public health care devices, HCI technologies, and other mechanisms. Even though some theoretical and experimental results have been generated, numerous issues still need to be addressed. The original feature selection, the specific choice of features, and the combination of features all need to be investigated further; in addition to the dimensionality reduction method, the feature selection method can be used to reduce the dimension, but the specific algorithm must be determined further.

Relying on Video frames and hand categorization masking, Benitez-Garcia et al. [39] created a highly effective and robust classifier for real-world usage. It strengthened the overall accuracy underlying two distinct HGR approaches, Temporal Segmentation Networking (TSN) but also Temporal Shifting Units, that used a lighter weight semantically segmented methodology (FASSD-Net) (TSM). The authors also evaluate the outcomes of both HGR approaches, finding that TSM recognizes motions that rely on temporal information, whereas TSN excels at static gestures. We intend to combine TSM and TSN in a single architecture for real-time recognition of both types of hand gestures in the future.

Tan et al. [40] created an improved densely linked DCNN network (EDenseNet), enabling vision-dependent hand motion identification. This redesigned transition layer in EDenseNet optimizes attribute propagation via leveraging image features. In addition, EDenseNet's redesigned transitioning layer enhances feature transmission through such a 1*1 bottleneck tier, whereas the Conv layer prunes and perhaps even smoothen out irrelevant attributes. Consequently, the overall count of trainable demanded parameters seems to be considerably diminished, leading to greater parameter effectiveness. Moreover, EDenseNet makes networks more prone to overfitting.

Mujahid et al. [41] developed a lightweight model for motion identification predicated mostly on YOLO v3 and DarkNet-53 ConvNet that does not require any extra pre-processing, picture filtering, or image augmentation. They created a system enabling the detection of hand gestures throughout real-time and motions via video sequence. However, many learning-based and image processing methodologies in gesture detection use pre-trained CNN concepts to extract the features. Implementing such an appropriate feature design strategy that incorporates hyperparameter adaptation, on the other hand, seems usually disregarded. Furthermore, the selection of hyper-parameter tweaking remains a serious challenge.

The current work highlights the two characteristics mentioned above, which serve as an incentive to determine the best-featured engineering and hyperparameter optimization approach toward enhanced gesture detection rate. The following section discusses the background of deep learning.

3. BACKGROUND

Deep learning emerges to be a network learning approach that has received much attention in artificial intelligence. Functioning with vast volumes of information, neural networks comprised of countless asymmetric hidden units containing neurons conduct retrieval of features, categorization, and pattern discovery, including conversion. With deep neural networks, all output information within one tier is the data input over the next layer. Deep learning does not require handling to retrieve dense characteristics from certain input information, which is essential for conventional learning methods. Post labelled information is often used as data input during deep learning. Data-dense properties were autonomously recovered but also taught utilizing diverse approaches in two or even more hidden units in neural networks through supervised/unsupervised properties learning, including hierarchical retrieval of features. The next section goes into great detail on capsule neural nets and how routing protocol is achieved in capsule neural nets.

3.1 Overview of capsule networks

A capsule is a collection of neurons whose activity vector

indicates the instantiation parameters of a particular entity, such as an object or an object constituent. To learn visual features like aspects, browsing, placement, scalability, and so on, Capsule Networks employ capsules made up of a group of neurons as well as a singular routing-by-agreement approach to satisfactorily recognize the numerous distinct angle norms of the same image that even the CNN was unable to recognize [42]. Several outputs of the capsule are grouped to create an activating vector. Each orientation of the activation vector provides basic exposure characteristics of the item, including such spot or even path. In contrast, the overall length of such activation vector (normal but rather magnitude) shows the predicted likelihood.

For instance, whenever we flip a picture, its activation vectors shift in length though not in width. These lengths of relatively low-level output capsules conform to their respective presence (e.g., eyes, nose, and mouth). Numerous properties of an item are encoded by vector dimensions, including size, direction, position, etc.

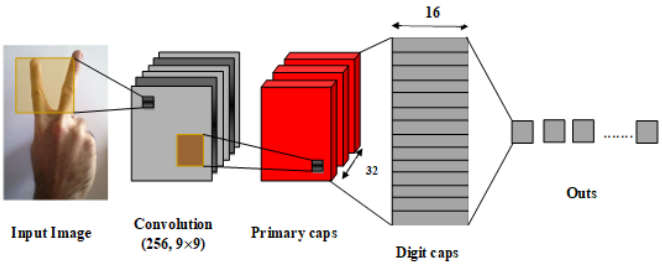

Figure 1. Capsule network sample model

After computing vectors in bottom-level capsules, the predicted estimates were guided towards the top-level capsules, which mostly precisely fit their forecasts, allowing the accessibility of items to be calculated more correctly using clearer data input. It would be referred to as dynamic routing. The overall topology of such capsule network structure is depicted in Figure 1. The same procedure of hand gesture identification utilizing an optimized capsule neural net has been presented in the forthcoming section.

4. RECOGNIZING HAND GESTURES BY OPTIMIZED CAPSULE NEURAL NETWORK

In deep learning research, high success rates have been achieved using CNN and capsule networks. While CNN has a high completion rate, its size depreciation and pixel shrinking employed for such a pooling layer are generally viewed as a drawback. Also, the performance of the network model will have an impact on modifying the hyperparameter to increase the accuracy of the detection of hand gestures. Furthermore, among the most basic issues underlying neural networks was a generalization. Thus to avoid this, our research introduces a Capsule Neural Network. A capsule is a group of neurons that learn to recognize objects or sections of objects in images. Unlike neurons, which produce scalars with no direction, capsules produce vectors with a direction. This characteristic in capsules aids in the resolution of the CNN orienting problem. When the picture's orientation is altered, the vector's direction shifts to match. It produces a vector whose length denotes the presence of an entity in an image. The vector's length serves as a confidence score. Longer vectors have a greater confidence score that the object exists in the image, whereas shorter vectors have a lower confidence value. The length of the vector instructs the network on which capsules to select for forwarding to a higher capsule where further processing will take place. The vector orientation specifies instantiation properties such as an object's rotation, size, or precise placement in an image. The capsules do extensive internal computations on the inputs to better encapsulate the findings by transforming the yields into a tiny vector of information outputs. Capsule networks were more successful than CNN architecture in images obtained from various perspectives.

Our proposed CapsNet seems to have the following architecture: Firstly, the Relu layer would be employed to train all data input, as well as the outputs have been used as data input for such capsule networks. The same neural net was again modified to enhance its reliability in identifying hand motions by inserting additional SoftMax layers before the output layer in the CapsNet architecture. Figure 2 depicts the proposed architecture model.

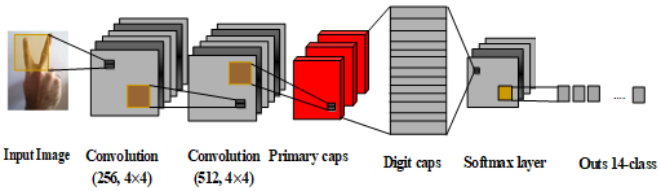

Figure 2. Proposed CapsNet architecture model

The input of 32×32 RGB images initially fed into our proposed optimized CapsNet architecture, as shown in Figure 2. The structure of a Capsule Neural Network comprises two convolution layers, primary caps, digits caps, and an additional SoftMax layer before the output class. The input image is transformed using a convolution layer to extract features from it, where the image is convolved with a filter in this transformation. In convolution layers, the hyperparameters are the size of the filter =4×4, stride = 1, padding = 0, and the activation function is a squashing function instead of Rectified Linear Unit (ReLU) are utilized to condense the information to a length of between 0 and 1. The activation function of the ReLU is also employed. This layer is in charge of turning pixel intensities into local feature detector activity. These outputs are subsequently supplied into the principal capsule layer as inputs.

Moreover, the inverse graphics process takes place at the primary capsule layer. This layer is in charge of capturing entities at the most basic level. A CapsNet seeks to borrow the concept of inverse graphics, which is the technique by which computers typically produce images in reverse. The network begins with a picture and attempts to discover the objects included inside it and instantiation factors such as posture (relative position, orientation, and size). In contrast, CNN disregard the spatial correlations between features.

To get a representation of instantiation parameters like,

1. We use a handful of convolution layers to generate an array of feature maps.
2. This array is reshaped to provide a collection of vectors for each position.
3. The final step is to ensure that no vector is longer than one. It is accomplished by using a squashing function to squash it so that its length is between 0 and 1.

The preservation of comprehensive information about an object's position and posture throughout the network is a major

characteristic of Capsule Networks. It is referred to as equivariance.

There are numerous pooling levels in CNNs. Researchers discovered that these pooling levels frequently lose information. Because Capsule Networks are equivariant, they may be used for object identification and picture segmentation applications.

The Matrix Multiplication is conducted on the input layer in the first step. We take a picture and turn it into vector values to grasp the spatial arrangement. Then, the weights of the inputs which way the current capsules should travel in the following layer. It operates in tandem with the Dynamic Routing Algorithm. The Dynamic Routing method is responsible for CapsNet's effectiveness. Its function is to provide communication between the Primary Capsule and the DigitCaps layer. A capsule in the bottom layer of the primary capsule layer must figure out how to deliver its output vector to the DigitCaps layer. This layer has one 6D capsule for each digit class. Every capsule receives input from capsules in the layer underneath it. The output of this layer is subsequently transmitted into the decoder network as input. The dynamic routing algorithm computes a coupling coefficient to characterize the relationship between the Primary Capsule and the DigitCaps layers. This coupling coefficient value is significant since it permits capsule routing to the proper following layers, which only agree with its inputs. This coefficient value, however, is not permanent. It has been updated. To optimize the loss quickly and minimize the problem of overfitting, only three routing iterations are proposed. It is how the network keeps learning by using a non-linear function to compress the data. Short vectors are squashed to virtually zero length, whereas large vectors are crushed to less than one length. Thus, the length of a capsule's output vector represents the chance that an entity is present in the current input. In a CapsNet, the squashing function is applied to a hierarchical group of layers rather than each layer as in CNNs.

Table 1. Hyperparameters values of Capsule Neural Network model

Kind of Layer	Size of filter	Stride	Filter count/capsule size	Padding	Activation
Convolutional	4×4	1	256	Same	ReLu
Convolutional	4×4	1	512	Same	ReLu
Primary Caps	4×4	2	8	Valid	ReLu
Digit Caps	4×4	2	16	Valid	ReLu
ReLu	4×4	1	512	Same	ReLu

The following hyper parameter values are used in the primary caps layer: The filter size is 4×4, its stride is 2, the digit's capsule is 8, the same channels comprise 32, and the squashing activation function is employed. The main caps layer comprises 32 basic capsules, one of which contains eight-dimensional vectors. A dynamic routing approach has been used to send capsule outputs among all high-levelled capsules inside the layer below. Eventually, sub-capsules for every potential high-levelled capsule have been multiplied by weight matrices to establish its output. Again, this output quantity is estimated using a non-linear "squash" mechanism that shortens the vectors to approximately zero length. Inside the category capsule tier, 8-dimensional units were turned into 16-dimensional matrices for every capsule via a weighted matrix and an encoding algorithm. Vector output data inside capsules substitute scalar output data inside a capsule. In addition, neural network models predict a multinomial probability distribution where the SoftMax function is utilized as an activation function in the output layer. Finally, the output is obtained from our proposed capsule neural network. An Adam optimizing method reduces the total error between the capsule neural net's actual output and the correct quantity. The rate of learning was set at roughly 0.001. This developed Capsule Neural Network model and hyperparameter values are shown in Table 1. Both frameworks, as well as parameters provided throughout this research, have been established after various evaluations.

The mathematical expression of the proposed model's functions is described below. Eq. (1) gives the mathematical expression for the convolution process.

$$b_k = g(\sum_j e_j f_j + c) \tag{1}$$

In Eq. (1), b_k seems to be the (j+1) tier output consequence; e_j have become the (kernel) weights first from the preceding layer; f_j is just the features mapping recovered from the preceding layer; c has been the biased value from the preceding layer, while g is now an activation function.

When such ReLU activation function is being employed in the conv layer, its output is 0 whenever that input score is less than zero, as well as the output becomes equal to its input quantity because once the input score is higher than 0, even a linear association with both the dependent factor is created. Eq. (2) shows the ReLU activation function represented in g(f) formula.

$$g(f) = \max(0, f) \tag{2}$$

At a depth of 8×32, the convolutional findings have been utilized again for the principal capsule. So every capsule has 32 channels and eight convolutional modules, along with 4×4 filters and one stride. These affine transformations have been done first from the scalar input data from the CNN network towards the capsule network's fundamental caps. Even the weights of such analysis were added and afterwards converted in vector-only with squash functions. A formula again for affine transformations has been provided in Eq. (3), as well as the equation of summing all weights is given in Eq. (4).

$$\hat{v}_{kmj} = e_{jk} v_j \tag{3}$$

$\widehat{v_{kmJ}}$ is the prediction vector derived in Eq. (3) by multiplying the weight matrix e_{jk} by the capsule output v_j.

$$t_j = \sum_j d_{jk} \hat{v}_{kmj} \tag{4}$$

In Eq. (4), t_j is indeed the total weight of either the predictive vectors, whereas d_{jk} would be the connection coefficients acquired either by iterative dynamical routing approach. The d_{jk} the formula is presented in Eq. (5).

$$d_{jk} = \frac{\exp(c_{jk})}{\sum_l \exp(c_{jl})} \tag{5}$$

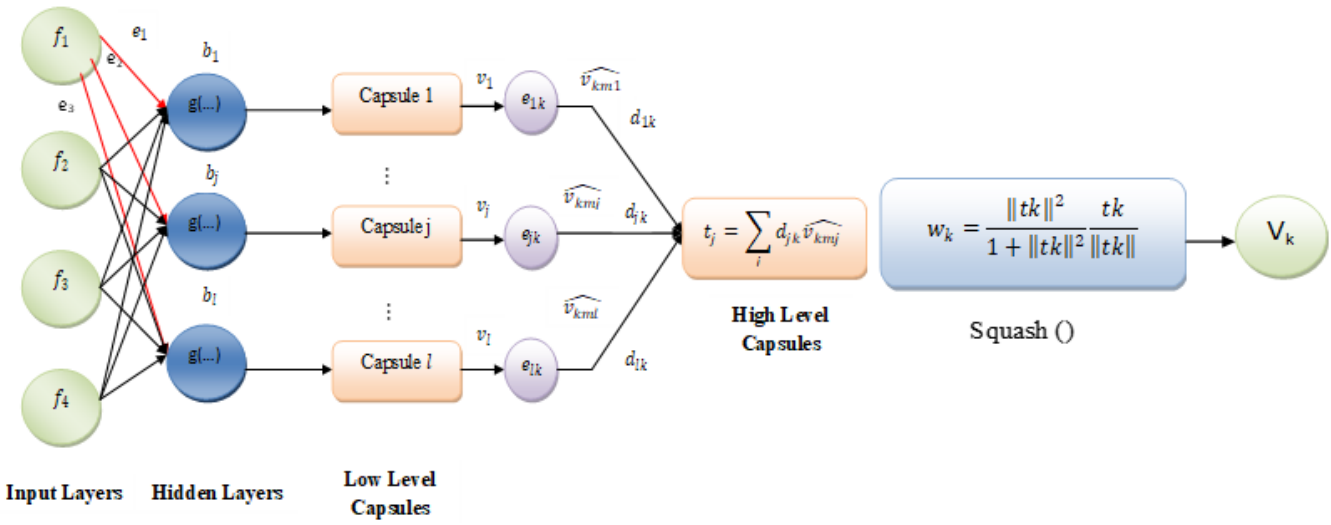

Figure 3. Mathematical expression of capsule neural network model

Even a non-linear squash function has been used to guarantee that perhaps the length of such a capsule's output vector equals zero whether it is short or somewhat lower than one if it is long. Eq. (6) contains the squash function formula.

$$w_k = \frac{tk^2}{1+tk^2}\frac{tk}{tk} \tag{6}$$

This same vector output for capsule k has been represented by w_k in Eq. (6). There are 16 capsules for each class in the output layer (DigitsCaps), relying just on the dataset's category count, and now those capsules acquire input data from either the preceding layers. At the capsule network's output, given into the SoftMax layer. At the capsule network's output, there are three Fully Connected layers. In contrast, the ReLU and sigmoid functions will be used towards fully linked layers that carry out the network's activation functionality. This sigmoid function has been utilized to recreate the picture. A sigmoid activation mechanism provides an output around zero for every component therein definition set. An expression for such a sigmoid activation function has been seen in Eq. (7).

$$g(f) = \frac{1}{(1+e^{-f})} \tag{7}$$

The full neural network architecture of such Capsule Neural Network model presented in this work is seen in Figure 3.

The image arrays are our model's inputs. A convolutional layer is even used to learn the feature mapping in the input image. Several dense features of such an image stream were developed utilizing multiple convolution kernels, as well as the weights were also obtained employing a CNN architecture. These weights were lowered, and indeed the findings are assessed for adequacy. Its scalar outcomes of the convolutional layer are always used as inputs towards the capsule net, having vector output capsules substituting those scalar outputs.

The Capsule Neural Network employs a high dynamic routing mechanism to ensure that its output attains its adequate capsule. Capsule outputs get passed among all high-levelled capsules inside the layer below. In contrast, sub-capsules for every conceivable high-levelled capsule were multiplied either by weight matrices or to compute output. This capsule network's output has been given through into the SoftMax layer. The output value is estimated even if the shorter vectors are shrunk toward a length nearly to 0, using a non-linear "squash".

5. RESULT AND DISCUSSION

This section goes through the implementation outcomes and the overall performance of our developed system. In addition, comparison results of existing works are presented.

Tool: PYTHON 3
OS: Windows 7 (64 bit)
Processor: Intel Premium
RAM: 8GB RAM

5.1 Dataset description

In our proposed work, we use two different types of datasets which are Human-Computer Interaction (HCI) [43] and Leap Motion hand gesture-based dataset [44].

5.1.1 Human-Computer Interaction (HCI) dataset

The Human-Computer Interaction (HCI) dynamic hand-gesture database seems to be a brand-new visual database established to test our hand-gesture detection methodology [43]. As illustrated in Figure 5, to conduct numerous mouse operations, distinct dynamic hand motions have been presented: pointer, left-clicking, right-clicking, mouse activating, and cursor deactivation. The collection of videos includes 30 training video sequences, each performed by different individuals. Each participant accomplishes 5 video sequences wherein they execute a different dynamic hand motion in numerous instances. Those 30 video sequences have been used to train the algorithm's duplicate detection and identification stages. In addition, the database comprises six extensive video sequences during testing. Every test video sequence exhibits a person's behaviour when utilizing the application and making distinct dynamic hand motions. Those test video sequences have been used to validate the whole system of hand motion detection.

5.1.2 Leap motion hand gesture-based dataset

The leap motion sensor captured several different dynamic motions in this dataset [44]. A collection of ten different motions were recorded from ten different people (5 women and five males). For each motion and participant, a total of 200 frames were recorded. Open palm parallel to the sensor (Palm), closed palm with the thumb and index fingers extended (L), palm closed (Fist), fist perpendicular to the sensor (Fist m), and palm closed with the thumb extended (Palm). A closed palm with the index extended (Index), an open palm with the index and thumb constituting a circle (OK), an open palm perpendicular to the sensor (Palm m), a semi-close palm in the shape of a 'C,' and an open palm with all its fingers separated (Palm m).

Figure 4 shows how the Human-Computer Interaction (HCI) dynamic hand-gesture database [43] images are pre-processed to create the threshold value. The pre-processed image's output is sent into the input of our new capsule neural network. Figure 4(a) depicts images taken from the HCI dynamic hand-gesture database, subsequently transformed into grayscale images, as seen in Figure 4(b). The threshold is then determined using local data such as the greyscale pixel area's mean, range, and variance. Figure 4(c) shows the thresholding after it has been constructed.

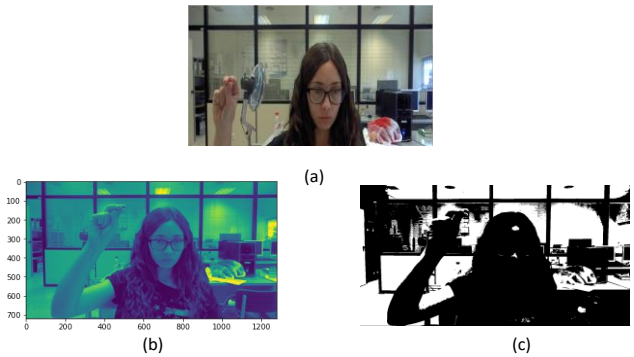

Figure 4. Processed images from database HCI

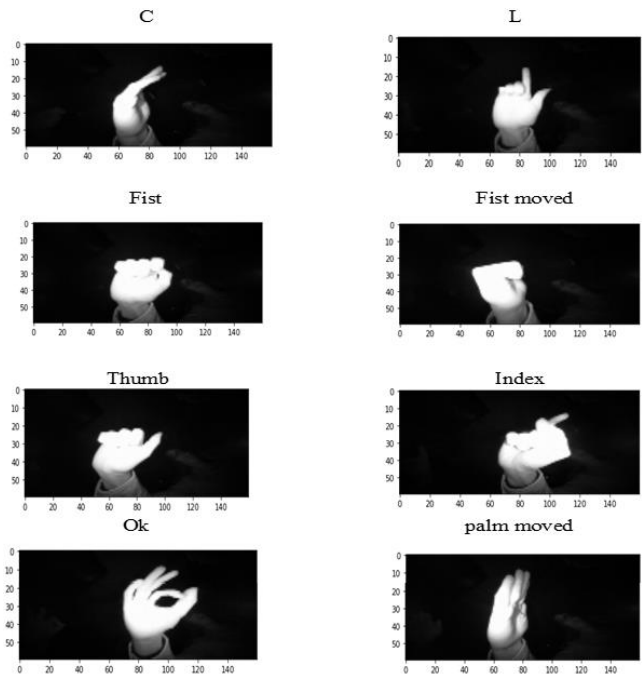

Figure 5. Images from the leap movement hand gesture-dependent dataset were utilized as input [44]

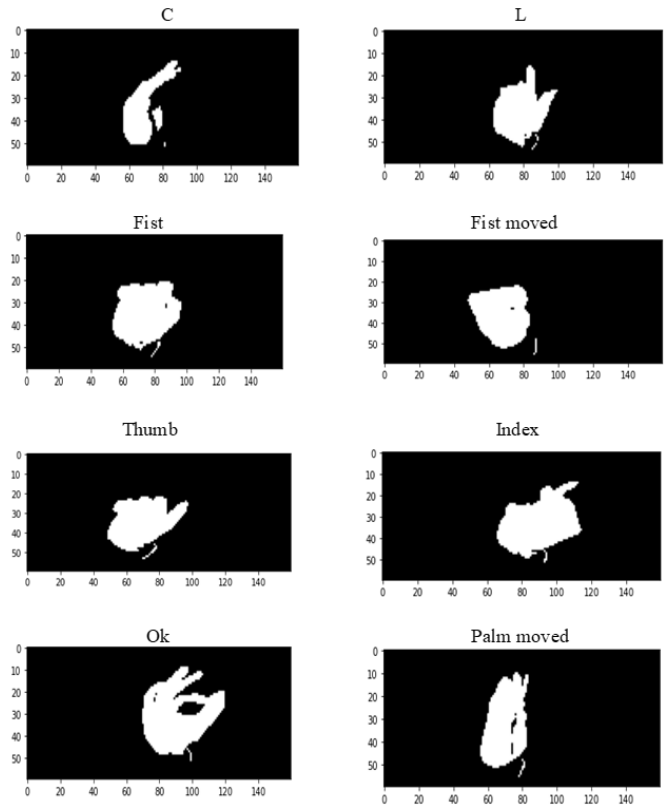

Figure 6. Normalized images from the given input

Figure 5 depicts the input images from the dataset in which dynamic hand gestures such as C, L, first, firstly moved, thumb, index, ok, and palm moved have been taken for this research work.

Figure 6 depicts the visuals normalized by our optimized capsule neural network. Normalization for information is executed by subtracting the mean out of each pixel and dividing this result by a standard deviation afterwards. While the network is being trained, this allows for rapid convergence. Furthermore, it overcomes the neural network's generalization difficulty.

5.2 Performance analysis

This section encompasses the overall performance assessment of the presented model, which again was assessed utilizing accuracy and loss measures.

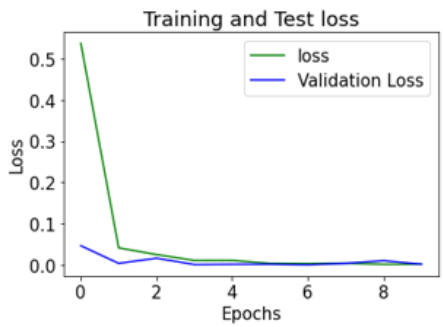

Figure 7. Training as well as test loss

5.2.1 Training as well as test loss

Training loss refers to a mistake in the training set of data. Validation loss pertains to the error after passing this same validation set of information via a trained network. Figure 7

depicts the proposed model's training and testing loss. Figure 7 shows the train loss graph, and the proposed optimized capsule neural network model achieved less than 0.1 train loss in two epochs. The proposed optimized capsule neural network model obtained less than 0.1 validation loss in three epochs.

5.2.2 Accuracy during training as well as testing

Training accuracy refers to the trained model's capacity to recognize independent pictures that were not utilized during training. The trained model's potential to detect independent pictures which were not utilized during training has been regarded as test accuracy. Figure 8 depicts the recommended model's training and testing accuracy. The presented optimized capsule neural network model obtained a learning rate of more than 95% in 2 epochs, as demonstrated in Figure 8, which is coloured green. The proposed optimized capsule neural network model obtained greater than 95% validation accuracy in three epochs, as indicated in the blue-coloured test accuracy graph.

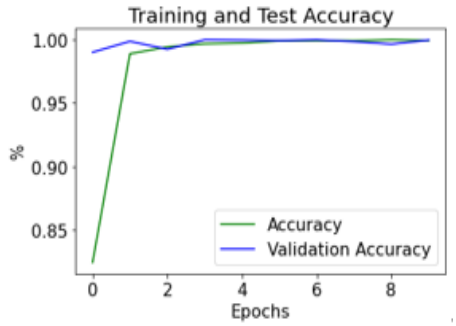

Figure 8. Accuracy during training as well as testing

5.2.3 Accuracy and loss

The average findings of overall tests conducted just on the HCI dataset have been visualized in Figure 9. While looking at Figure 9, it is clear that the recommended optimized capsule neural network model outperformed the others in the testing. The presented optimized capsule neural network model obtained a more than 90% learning rate in two epochs. Figure 9 shows that the proposed optimized capsule neural network model achieved less than 10% train loss in two epochs.

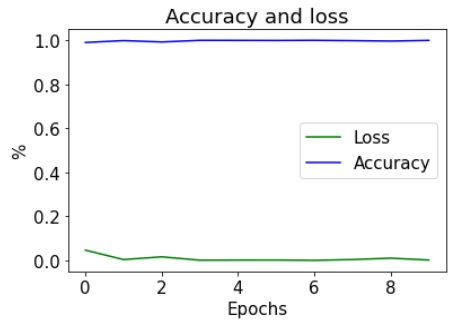

Figure 9. Accuracy and loss

5.2.4 Error rate

The erroneous rate was specified simply as the fraction of false data units to the total amount of data units transferred. Figure 10 depicts the error rate of a proposed technique vs the number of epochs utilizing optimal capsule neural network model training and testing data. Our advanced technique's performance is tested, and the error rate decreases as epochs grow. Furthermore, Figure 10 shows that the error rate stays constant after ten epochs.

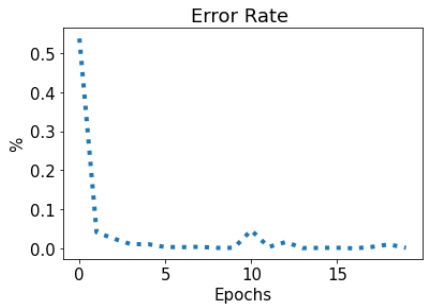

Figure 10. Error rate

5.2.5 Execution time

Figure 11 displays the developed model's epoch-depended operational time. The overall effectiveness of our presented strategy was tested, and as the number of epochs rises, the execution time decreases. As a result of the innovative, optimized capsule neural network model, our execution time is reduced.

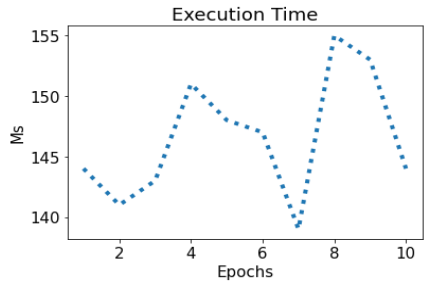

Figure 11. Execution time with epochs

5.2.6 Confusion matrix

Figure 12 exhibits typical confusion matrix underlying hand gesture identification. The proposed optimized capsule neural network produces the confusion matrix, in which these same diagonal values symbolize the overall count of effectively categorized tuples through the algorithms. In contrast, the off-diagonal results depict the count of misclassified tuples even by designs. As higher the diagonal value, the better its effectiveness.

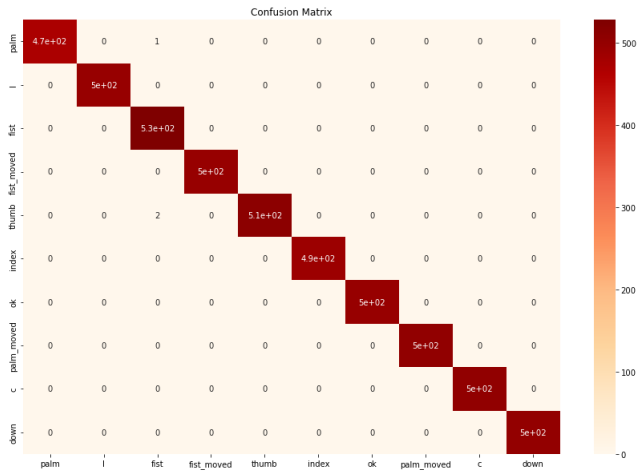

Figure 12. Confusion matrix

5.3 Comparison results

This section describes the proposed technique's comparison results, in which our novel technique is compared to baseline approaches such as volumetric Spatiograms of either the Local Binary Pattern (VS-LBP) [45], Local Binary Pattern (LBP) [46], Temporal Pyramid Matching of the Local Binary Pattern (TPM-LBP) [47], Pyramid Histogram of Gradients (PHOG) [48], as well as Scale Invariant Feature Transform (SIFT) [49].

Table 2. Overall accuracy

Methods	Accuracy (%)
VS-LBP [45]	92.7
LBP [46]	91.5
TPM-LBP [47]	96.5
PHOG [48]	94.6
SIFT [49]	97.6
Proposed Method	99.5

Figure 13 depicts the total accuracy comparison. The suggested approach achieves improved accuracy by using an optimized capsule neural network. Our proposed approach outperformed the baseline Volumetric Spatiograms of Local Binary Pattern (VS-LBP) [45], Local Binary Pattern (LBP) [46], Temporal Pyramid Matching of Local Binary Pattern (TPM-LBP) [47], Pyramid Histogram of Gradients (PHOG) [47], as well as Scale Invariant Feature Transform (SIFT) [49] by 92.7, 91.5, 96.5, 94.6, and 97.6% and tabulated in Table 2. As a result, our unique approach has a greater accuracy of 99.5% than existing techniques.

Figure 14 depicts the total error rate comparison. An improved capsule neural network reduces the error rate of the proposed technique. Our suggested method outperformed the baseline Hidden Markov Model (HMM) [50], Hand Movements Temporal Features (HMTF) [51], Pyramid Histogram of Gradients (PHOG Top) [47], and Scale Invariant Feature Transform (SIFT) [49] by 35.7%, 27.6%, 11%, and 7%, respectively, which is tabulated in Table 3. As a result, our unique approach has a 5% error rate, greater than the existing strategies.

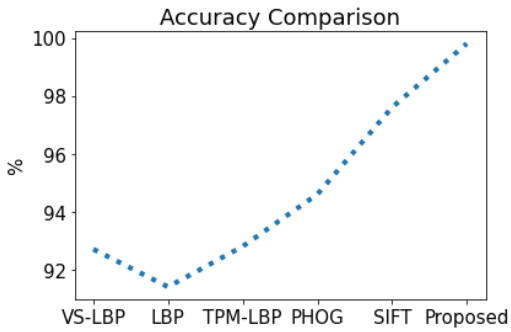

Figure 13. Overall accuracy

Table 3. Overall error rate

Methods	Error Rate (%)
HMM [50]	35.7
HMTF [51]	27.6
PHOG_Top [48]	11
SIFT [49]	7
Proposed Method	5

Figure 14. Overall error rate

Table 4. Accuracy comparison

Methods	Accuracy (%)
Deep LSTM [52]	86.18
HBU-LSTM [52]	89.98
CNN-SVM [53]	97.28
Proposed Method	99.5

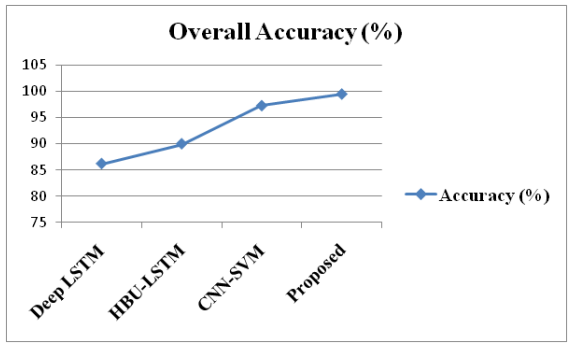

Figure 15. Accuracy comparison

Figure 15 depicts the total accuracy comparison. The proposed approach attains improved accuracy by using an optimized capsule neural network. Our proposed approach outperformed the baseline Deep Long Short Term Memory (D-LSTM) [52], Hybrid Bidirectional Uni-directional Long Short Term Memory (HBU-LSTM) [52], and Convolutional Neural Network with Support Vector Machine (CNN-SVM) [53] such as 86.18%, 89.98%, and 97.28% which is accumulated in Table 4. As a result, our unique approach has a greater accuracy of 99.5% than existing techniques.

We may infer from the visual assessment above that the developed strategy surpasses some conventional approaches regarding identification accuracy, rate of error, and performance.

6. CONCLUSION

Hand gesture detection has been the most effective communication tool in human-computer interaction, with a broad range of purposes. Deep learning hand gesture detection work has utilized CNN, RNN, and LSTM, including 3D CNN concepts with intelligent automobile control, sign language identification systems, wearable technologies, robotic devices, and virtual reality uses. While assessed jointly, deep learning systems utilized with hand gesture detection were highly effective throughout all tests. In contrast to this research, we proposed a novel model called an optimized capsule neural

network. Thus, by removing the pooling layer, our suggested model minimizes the computational complexity of the neural network and outperforms existing neural network models. When the training results were compared, the developed hybrid system had a maximum accuracy rate of 99.5% in these hand gesture-dependent datasets.

REFERENCES

[1] Chaudhary, A., Raheja, J.L., Das, K., Raheja, S. (2013). Intelligent approach to interact with machines naturally using hand gesture recognition: Survey. arxiv: 1303.2292. https://doi.org/10.48550/arXiv.1303.2292

[2] McIntosh, J., Marzo, A., Fraser, M. (2017). Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology, pp. 593-597. https://doi.org/10.1145/3126594.3126604

[3] Pisharady, P.K., Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. Computer Vision and Image Understanding, 141: 152-165. https://doi.org/10.1016/j.cviu.2015.08.004

[4] Fang, Y., Wang, K., Cheng, J., Lu, H. (2007). A real-time hand gesture recognition method. In 2007 IEEE International Conference on Multimedia and Expo, pp. 995-998. https://doi.org/10.1109/ICME.2007.4284820

[5] Oudah, M., Al-Naji, A., Chahl, J. (2020). Hand gesture recognition based on computer vision: A review of techniques. Journal of Imaging, 6(8): 73. https://doi.org/10.3390/jimaging6080073

[6] Fang, Z. (1999). Computer gesture input and its application in human computer interaction. Minimicro Systems-Shenyang, 20: 418-421.

[7] 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. https://doi.org/10.1109/TSMCC.2007.893280

[8] Ahuja, M.K., Singh, A. (2015). Static vision based Hand Gesture recognition using principal component analysis. In 2015 IEEE 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE), pp. 402-406. https://doi.org/10.1109/MITE.2015.7375353

[9] Kramer, R.K., Majidi, C., Sahai, R., Wood, R.J. (2011). Soft curvature sensors for joint angle proprioception. In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1919-1926. https://doi.org/10.1109/IROS.2011.6094701

[10] Jesperson, E., Neuman, M.R. (1988). A thin film strain gauge angular displacement sensor for measuring finger joint angles. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, p. 807. https://doi.org/10.1109/IEMBS.1988.95058

[11] Fujiwara, E., dos Santos, M.F.M., Suzuki, C.K. (2014). Flexible optical fiber bending transducer for application in glove-based sensors. IEEE Sensors Journal, 14(10): 3631-3636. https://doi.org/10.1109/JSEN.2014.2330998

[12] Shrote, S.B., Deshpande, M., Deshmukh, P., Mathapati, S. (2014). Assistive Translator for Deaf & Dumb People. International Journal of Electronics Communication and Computer Engineering, 5(4): 86-89.

[13] Gupta, H.P., Chudgar, H.S., Mukherjee, S., Dutta, T., Sharma, K. (2016). A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. IEEE Sensors Journal, 16(16): 6425-6432. https://doi.org/10.1109/JSEN.2016.2581023

[14] Al-Hammadi, M., Muhammad, G., Abdul, W., Alsulaiman, M., Bencherif, M.A., Alrayes, T.S., Mekhtiche, M.A. (2020). Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. IEEE Access, 8: 192527-192542. https://doi.org/10.1109/ACCESS.2020.3032140

[15] Vaitkevičius, A., Taroza, M., Blažauskas, T., Damaševičius, R., Maskeliūnas, R., Woźniak, M. (2019). Recognition of American sign language gestures in a virtual reality using leap motion. Applied Sciences, 9(3): 445. https://doi.org/10.3390/app9030445

[16] Rezende, T.M., Almeida, S.G.M., Guimarães, F.G. (2021). Development and validation of a Brazilian sign language database for human gesture recognition. Neural Computing and Applications, 33(16): 10449-10467. https://doi.org/10.1007/s00521-021-05802-4

[17] Žemgulys, J., Raudonis, V., Maskeliūnas, R., Damaševičius, R. (2020). Recognition of basketball referee signals from real-time videos. Journal of Ambient Intelligence and Humanized Computing, 11(3): 979-991. https://doi.org/10.1007/s12652-019-01209-1

[18] Afza, F., Khan, M.A., Sharif, M., Kadry, S., Manogaran, G., Saba, T., Damaševičius, R. (2021). A framework of human action recognition using length control features fusion and weighted entropy-variances based feature selection. Image and Vision Computing, 106: 104090. https://doi.org/10.1016/j.imavis.2020.104090

[19] Nikolaidis, A., Pitas, I. (2000). Facial feature extraction and pose determination. Pattern Recognition, 33(11): 1783-1791. https://doi.org/10.1016/S0031-3203(99)00176-4

[20] Kulikajevas, A., Maskeliunas, R., Damaševičius, R. (2021). Detection of sitting posture using hierarchical image composition and deep learning. PeerJ Computer Science, 7: e442. https://doi.org/10.7717/peerj-cs.442

[21] Ryselis, K., Petkus, T., Blažauskas, T., Maskeliūnas, R., Damaševičius, R. (2020). Multiple Kinect based system to monitor and analyze key performance indicators of physical training. Human-Centric Computing and Information Sciences, 10(1): 1-22. https://doi.org/10.1186/s13673-020-00256-4

[22] Huu, P.N., Minh, Q.T. (2020). An ANN-based gesture recognition algorithm for smart-home applications. KSII Transactions on Internet and Information Systems (TIIS), 14(5): 1967-1983. https://doi.org/10.3837/tiis.2020.05.006

[23] Kour, K.P., Mathew, L. (2017). Sign language recognition using image processing. International Journal of Advanced Research in Computer Science and Software Engineering, 7(8): 10.

[24] Kumar, B.P., Manjunatha, M.B. (2017). A hybrid gesture recognition method for American sign language. Indian Journal of Science and Technology, 10(1): 1-12. https://doi.org/10.17485/ijst/2017/v10i1/109389

[25] Muthukumar, K., Poorani, S., Gobhinath, S. (2017). Vision based hand gesture recognition for Indian sign languages using local binary patterns with support vector machine classifier. Advances in Natural and Applied

Sciences, 11(6): 314-322.
[26] Hu, Y. (2018). Finger spelling recognition using depth information and support vector machine. Multimedia Tools and Applications, 77(21): 29043-29057. https://doi.org/10.1007/s11042-018-6102-6
[27] Jadooki, S., Mohamad, D., Saba, T., Almazyad, A.S., Rehman, A. (2017). Fused features mining for depth-based hand gesture recognition to classify blind human communication. Neural Computing and Applications, 28(11): 3285-3294. https://doi.org/10.1007/s00521-016-2244-5
[28] Nakjai, P., Katanyukul, T. (2019). Hand sign recognition for Thai finger spelling: An application of convolution neural network. Journal of Signal Processing Systems, 91(2): 131-146. https://doi.org/10.1007/s11265-018-1375-6
[29] Perimal, M., Basah, S.N., Safar, M.J.A., Yazid, H. (2018). Hand-gesture recognition-algorithm based on finger counting. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-13): 19-24.
[30] Chen, N., Chen, Y.P., Wang, Q.F., Wu, S.P., Zhang, H.Y. (2022). MAF-DeepLab: A multiscale attention fusion network for semantic segmentation. Traitement du Signal, 39(2): 407-417. https://doi.org/10.18280/ts.390202
[31] Li, Y., Wang, X., Liu, W., Feng, B. (2018). Deep attention network for joint hand gesture localization and recognition using static RGB-D images. Information Sciences, 441: 66-78. https://doi.org/10.1016/j.ins.2018.02.024
[32] Alani, A.A., Cosma, G., Taherkhani, A., McGinnity, T. M. (2018, May). Hand gesture recognition using an adapted convolutional neural network with data augmentation. In 2018 4th International Conference on Information Management (ICIM), pp. 5-12. https://doi.org/10.1109/INFOMAN.2018.8392660
[33] Nunez, J.C., Cabido, R., Pantrigo, J.J., Montemayor, A.S., Velez, J.F. (2018). Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition, 76: 80-94. https://doi.org/10.1016/j.patcog.2017.10.033
[34] Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G., Jiang, D., Liu, H. (2019). Hand gesture recognition based on convolution neural network. Cluster Computing, 22(2): 2719-2729. https://doi.org/10.1007/s10586-017-1435-x
[35] Cheng, W., Sun, Y., Li, G., Jiang, G., Liu, H. (2019). Jointly network: a network based on CNN and RBM for gesture recognition. Neural Computing and Applications, 31(1): 309-323. https://doi.org/10.1007/s00521-018-3775-8
[36] Skaria, S., Al-Hourani, A., Lech, M., Evans, R.J. (2019). Hand-gesture recognition using two-antenna Doppler radar with deep convolutional neural networks. IEEE Sensors Journal, 19(8): 3041-3048. https://doi.org/10.1109/JSEN.2019.2892073
[37] Wu, X.Y. (2020). A hand gesture recognition algorithm based on DC-CNN. Multimedia Tools and Applications, 79(13): 9193-9205. https://doi.org/10.1007/s11042-019-7193-4
[38] Qi, J., Jiang, G., Li, G., Sun, Y., Tao, B. (2020). Surface EMG hand gesture recognition system based on PCA and GRNN. Neural Computing and Applications, 32(10): 6343-6351. https://doi.org/10.1007/s00521-019-04142-8
[39] Benitez-Garcia, G., Prudente-Tixteco, L., Castro-Madrid, L.C., Toscano-Medina, R., Olivares-Mercado, J., Sanchez-Perez, G., Villalba, L.J.G. (2021). Improving real-time hand gesture recognition with semantic segmentation. Sensors, 21(2): 356. https://doi.org/10.3390/s21020356
[40] Tan, Y.S., Lim, K.M., Lee, C.P. (2021). Hand gesture recognition via enhanced densely connected convolutional neural network. Expert Systems with Applications, 175: 114797. https://doi.org/10.1016/j.eswa.2021.114797
[41] Mujahid, A., Awan, M.J., Yasin, A., Mohammed, M.A., Damaševičius, R., Maskeliūnas, R., Abdulkareem, K.H. (2021). Real-time hand gesture recognition based on deep learning YOLOv3 model. Applied Sciences, 11(9): 4164. https://doi.org/10.3390/app11094164
[42] Vijayakumar, T. (2019). Comparative study of capsule neural network in various applications. Journal of Artificial Intelligence, 1(1): 19-27. https://doi.org/10.36548/jaicn.2019.1.003
[43] Maqueda, A.I., del-Blanco, C.R., Jaureguizar, F., García, N. (2015). Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. Computer Vision and Image Understanding, 141: 126-137. https://doi.org/10.1016/j.cviu.2015.07.009
[44] Mantecón, T., del-Blanco, C.R., Jaureguizar, F., García, N. (2016). Hand gesture recognition using infrared imagery provided by leap motion controller. In International Conference on Advanced Concepts for Intelligent Vision Systems, pp. 47-57. https://doi.org/10.1007/978-3-319-48680-2_5
[45] Ana Carlos, R. (2015). Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. Computer Vision and Image Understanding, Special Issue on Posture & Gesture, 141: 126-137.
[46] Al-Berry, M.N., Salem, M.A.M., Ebeid, H.M., Hussein, A.S., Tolba, M.F. (2016). Fusing directional wavelet local binary pattern and moments for human action recognition. IET Computer Vision, 10(2): 153-162. https://doi.org/10.1049/iet-cvi.2015.0087
[47] Maqueda, A.I., del-Blanco, C.R., Jaureguizar, F., García, N. (2016). Temporal pyramid matching of local binary subpatterns for hand-gesture recognition. IEEE Signal Processing Letters, 23(8): 1037-1041. https://doi.org/10.1109/LSP.2016.2579664
[48] Suni, S.S., Gopakumar, K. (2019). Fusing Multimodal features for Recognizing Hand Gestures. In 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), pp. 1-6. https://doi.org/10.1109/ICACC2019.8882910
[49] Park, S.K., Chung, J.H., Kang, T.K., Lim, M.T. (2021). Binary dense sift flow based two stream CNN for human action recognition. Multimedia Tools and Applications, 80(28): 35697-35720. https://doi.org/10.1007/s11042-021-10795-2
[50] Ahmed, A., Riaz, M.T., Raza, A., Zaib, A., Akbar, M.A., Sarwar, M.B. (2019). Modeling and simulation of office desk illumination using ZEMAX. In 2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), pp. 1-6. https://doi.org/10.1109/ICECCE47252.2019.8940756

[51] Abdalla, M.S., Hemayed, E.E. (2013). Dynamic hand gesture recognition of Arabic sign language using hand motion trajectory features. Global Journal of Computer Science and Technology.
[52] Ameur, S., Khalifa, A.B., Bouhlel, M.S. (2020). A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion. Entertainment Computing, 35: 100373. https://doi.org/10.1016/j.entcom.2020.100373
[53] Rahim, M.A., Islam, M.R., Shin, J. (2019). Non-touch sign word recognition based on dynamic hand gesture using hybrid segmentation and CNN feature fusion. Applied Sciences, 9(18): 3790. https://doi.org/10.3390/app9183790