Multimedia Tools and Applications (2024) 83:10587–10613
https://doi.org/10.1007/s11042-023-15865-1

mIV3Net: modified inception V3 network for hand gesture recognition

Bhumika Karsh[1] · **R. H. Laskar**[1] · **R. K. Karsh**[1]

Received: 16 September 2022 / Revised: 6 March 2023 / Accepted: 15 May 2023 /
Published online: 23 June 2023

Abstract
Hand gesture plays an important role in communication among the hearing and speech disorders people. Hand gesture recognition (HGR) is the backbone of human–computer interaction (HCI). Most of the reported hand gesture recognition techniques suffer due to the complex backgrounds. As per the literature, most of the existing HGR methods have only selected a few inter-class similar gestures for recognition performance. This paper proposes a two-phase deep learning-based HGR system to mitigate the complex background issue and consider all gesture classes. In the first phase, inception V3 architecture is improved and named mIV3Net: modified inception V3 network to reduce the computational resource requirement. In the second phase, mIV3Net has been fine-tuned to offer more attention to prominent features. As a result, better abstract knowledge has been used for gesture recognition. Hence, the proposed algorithm has more discrimination characteristics. The efficacy of the proposed two-phase-based HGR system is validated and generalized through experimentation using five publicly available standard datasets: MUGD, ISL, ArSL, NUS-I, and NUS-II. The accuracy values of the proposed system on five datasets in the above order are 97.14%, 99.3%, 97.4%, 99%, and 99.8%, which indicates significant improvement, i.e., 12.58%, 2.54%, 2.73%, 0.56%, and 2.02%, respectively, than the state-of-the-art HGR systems.

Keywords Hand gesture recognition (HGR) · Inception V3 · Sign language recognition · Deep learning · Transfer learning · Human–computer interaction (HCI)

✉ Bhumika Karsh
bhumika21_rs@ece.nits.ac.in

R. H. Laskar
rhlaskar@ece.nits.ac.in

R. K. Karsh
ram@ece.nits.ac.in

[1] Speech and Image Processing Laboratory, Department of ECE, NIT Silchar, Silchar 788010, Assam, India

1 Introduction

Computers significantly impact our daily lives due to the advancement of technology and their affordable low cost [2]. Hand gestures have been used as a communication language for hearing and speech disorders people. Worldwide, 430 million people have hearing loss, as per the world health organization (WHO). The researchers have predicted that this number may rise to 700 million by 2050 [3]. The “deaf” or “hard of hearing” can use only sign language in their communications among themselves and with others. Different countries have their own sign language. The researchers have paid large attention to helping hearing and speech disorders people by evolving automatic sign language recognition (ASLR) algorithms. Besides, HGR has been widely used for other applications, i.e., designing touchless systems, robotics devices [38], medical applications, and smart environments [10], etc. The primary goal of designing an ASLR algorithm is the conversion of hand gestures into "voice" or "text" with better accuracy and less computational cost. In other words, the main goal of the HGR is to classify and identify hand gestures correctly. Several methods and concepts from several fields, including image processing and neural network, have been used in the hand gesture recognition methodology to learn various hand postures. The end goal of reliable HGR is high recognition accuracy. The evolution of convolution neural networks, especially deep neural networks, excel in recognizing complicated patterns with complex backgrounds.

To help "deaf" or "hard of hearing", designing a robust hand gesture recognition system is an essential component for sign language interpretation. The population with hearing loss has a noticeable communication gap. A translator that converts gestures into verbal language can overcome this communication gap. A translator based on a robust HGR can help the hearing-loss population, which allows them to more easily and independently integrate into society.

Generally, hand gestures are detected and recognized based on two approaches, i.e., “sensors mounted on the hand [16, 34, 52]” and “using a photography camera called vision-based sensors [15]”. The second approach has been considered better than the earlier one, because hands are free from sensors to perform gestures [48]. The second approach has again been divided into two categories, i.e., static and dynamic [47]. In static gesture recognition, the feature extraction approach has a key pre-processing role in the pattern recognition problem. In the traditional pattern recognition procedure, feature extraction is a vital step. The prominent selected features are responsible for discriminating the hand gestures into different classes. It is a very challenging task to recognize gestures from a complex background. There are some approaches for gesture recognition in complex backgrounds [21, 54, 57]. Zhang et al. [54] presented HGR based on a hand pose estimator. In this method, the performance has been evaluated only in the indoor scenarios. The estimation of joints may be limited in the occlusion and low illumination scenarios. Li et al. [21] employed YCbCr color space for the segmentation of hands. In another work, Zhou et al. [57] proposed a network for segmentation based on dilated residual network and decoder. The performances of the methods [21, 57] are limited due to the skin color background.

On the other hand, dynamic gestures [27, 33, 56] include head movements for “No” and “Yes” that can be predicted using only temporal context data. In this paper, we have focused on the static HGR system because the deaf typically expresses the alphabet and digits using hand poses and fingers. Most of the conventional feature extraction methods are unable to detect some important salient features for distinguishing inter-class similar gestures. As a result, most of the existing methods have selected a couple of basic discriminating gestures for recognition. In this paper, all gesture classes given in five publicly

available datasets from various countries have been considered for recognition performance. Recently, researchers have paid large attention to deep learning for HGR [1, 3, 7, 17, 19, 21, 25, 31, 37, 49, 53, 54, 57]. Most of the existing approaches suffer from complex backgrounds and inter-class similarities.

The main contributions are as follows. 1. It has been observed from the literature that most of the existing HGR systems suffer from complex backgrounds and inter-class similarities. This paper proposes a two-phase deep learning-based HGR system to mitigate the complex background issue and consider all gesture classes. In the first phase, inception V3 architecture is improved and named mIV3Net to reduce the computational resource requirement. In the second phase, mIV3Net has been fine-tuned to offer more attention to prominent features. As a result, better abstract knowledge has been used for gesture recognition. 2. For generalizations, mIV3Net has been tested on five different hand gesture datasets, i.e., MUGD, ISL, ArSL, NUS-I, and NUS-II, using two different validation strategies. 3. Different transfer learning approaches have also been investigated along with the proposed one. 4. To show the efficiency, an optimal model is extracted by varying hyperparameters, i.e., learning rates, the number of epochs, batch size, and dropout rate. 5. This work also presents the prediction accuracy and analysis of each character of the five sign language datasets.

The remaining part of the paper has been structured as follows. Section 2 presents the recent related literature. The research gap is discussed in Section 3. The proposed methodology for HGR using modified inception V3 is discussed in Section 4. Section 5 provides the experimental results and their discussions. Finally, the conclusion and future scopes have been shown in Section 6.

2 Related work

Gesture recognition (GR) algorithms may be segregated into two categories based on gesture acquisition [30]. In the first category, GR uses sensors that mount on the hand. This approach uses sensor-equipped electronic hand globes to collect gesture data, which can be processed further for investigations and classifications [16, 24, 43, 52]. This category has better robustness and accuracy, but has a limited range of applications due to the need for specialized equipment. The second category of GR is called vision-based methods, in which the first step is acquiring images via camera. The acquired images are then passed through various operations of an image processing for gesture recognition [31, 40]. This category has received considerable attention from researchers due to the relatively least requirements of specialized equipment. In the second category of GR, hand gestures are recognized and classified using traditional hand-crafted features [9, 28] and recent deep learning architectures [1, 6–8, 14, 17, 19, 21–23, 25, 26, 29, 31, 32, 35, 37, 39, 41, 45, 49, 51, 53–55, 57].

An edge-oriented histogram was used by Nagarajan et al. [28] to detect static gestures. The authors derived features from the histogram. This method has an overall accuracy of 93.75%. A super pixel-based HGR system was introduced by Wang et al. [50]. This approach is based on combining a Kinect depth camera with a unique superpixel earth mover's distance metric. Here, markerless hand extraction is created by effectively utilizing Kinect's depth and skeletal data. The performance accuracy of this approach is 75.8% and 99.6% on two open datasets. Gupta et al. [9] introduced the combined properties of the SIFT and HOG to recognize gestures. The classification process employed a typical KNN classifier. In this approach, some sample gestures from the dataset have been selected for

experimental evaluation. It has been observed from the literature that traditional feature extraction methods may overlook some of the important features during classification.

The HGR system based on deep learning, in the vision-based category, has received a lot of attention. Since deep learning is better at extracting features and taking advantage of current advancements in computing. The HGR system using CNN was presented by Lin et al. [25]. The images in the dataset were registered using an Xbox Kinect camera. The authors attained 95.96% recognition accuracy. Li et al. [22] used new feature learning technology for identifying gestures, including the sparse auto-encoder. This technique is built for RGB-D images using principal component analysis and a sparse auto-encoder. This approach has 99.05% recognition accuracy. Further, Oyedotun et al. [31] included complex neural networks with lower error rates. The authors recognized the whole set of 24 hand gestures from Moeslund's database using deep learning. The maximum recognition rate attained was 92.83%.

Li et al. [23] established a method for training CNN via soft consideration approach using RGB-D images. A global sum is generated to represent the entire image, focusing mostly on the relevant weights. This method has attained accuracy values of 98.5% and 73.4%. Ranga et al. [37] have employed conventional feature extraction methods and convolutional neural networks for the recognition challenge. The authors evaluated the performance of various classifiers. The authors reported 97.01% accuracy. Chevtchenko et al. [7] have combined traditional and deep learning-based features. An approach for the optimization of hyperparameters was also recommended by Ozcan et al. [32]. The authors tested their approach using sign language digits and the Thomas Moeslund dataset. An accuracy of 98.09% was achieved on the Thomas Moeslund dataset.

Neethu et al. [29] attempted to recognize hand gestures using finger detection with CNN. Wadhawan et al. [49] performance evaluation of CNN on sign language recognition uses various optimizers. The authors assessed the performance of the system using different CNN models. According to experimental analysis, characteristics like the number of filters and layers have varied to reach the best level of validation accuracy. Furthermore, Liu et al. [26] proposed 19 layers of CNN for classifying and identifying hand gestures. This method has reported 99.2%. HGR system based on two-stage reported in [8]. For the job of segmenting and recognizing hand gestures, the authors took into consideration two stages. In the second stage, the segmented data and RGB information are combined for classification. An F-score of 88.10% has been revealed via experimental examination. Rathi et al. [39] created two-level architecture to classify and estimate the gesture classes. On a total of 12,048 test images, this method has an accuracy of 99.03%. However, the use of RGB-D data is a major limitation that requires a specialized depth sensor. The HGR systems designed in [6, 14, 35, 41, 45, 51, 53, 55] have used CNNs without modifying the structure, i.e., how many layers may be good enough for the task at the hand. Most of the methods are tested for some specific gesture classes. State-of-the-art deep CNN designs are computationally expensive and require a lot of labelled data during training. The literature has noted that traditional feature extraction methods overlook crucial features to differentiate between classes of similar gestures. Simple discriminating gestures for recognition have been taken into account by the majority of existing methods.

In this paper, we have proposed a two-phase deep learning-based HGR system to mitigate the complex background issue and considered all gesture classes. In the first phase, inception V3 architecture is modified and named mIV3Net to reduce the computational resource requirement. In the second phase, mIV3Net has been fine-tuned to offer more attention to prominent features. As a result, better abstract knowledge has been used for gesture recognition. Hence, the proposed algorithm has more discrimination characteristics and achieves better accuracy than the existing related methods.

3 Research gap

According to an assessment of the literature, hand gesture recognition has obtained considerable success using conventional CNN-based methods like ResNet-50 [45], DenseNet-121 [43], MobileNet [12], etc. However, these deep neural networks require computational resources. Besides, the issue of gradient and negative learning is the major obstacle that deep architectures must overcome. In these networks, the same modules are repeatedly stacked, which cause an over-adaption of hyperparameters for certain issues. Due to their sophisticated structures, these networks can be modified and used on platforms with limited time and processing resources. Therefore, we proposed a mIV3Net that reduces the repeatedly stacked approach by empirically selecting necessary layers and thus requiring less computational resources. mIV3Net has been fine-tuned to offer more attention to prominent features. Furthermore, depending on training data, most of the CNNs have an issue of overfitting and have lower accuracy. The mIV3Net network's fundamental structure allows it to overcome this issue.

4 Proposed methodology

This section discusses the overall HGR system, the architecture of the proposed mIV3Net, and fine-tuning of mIV3Net. The overall HGR system is shown in Fig. 1, where the process of the system has been separated into three stages. In the first stage, the gestures from selected datasets are prepared to give input into mIV3Net, called pre-processing. Next, the features are extracted using mIV3Net. Finally, the extracted features are fed into the classifier to segregate the gestures into the corresponding classes. Each stage of Fig. 1 has been discussed in the following subsection.

4.1 Pre-processing

The HGR system was tested using five different countries' sign language datasets. The images in the datasets have various dimensions with variations in geometry. In our case, we have used the CNN model, i.e., mIV3Net, for the feature extractions. Since mIV3Net demands fixed-size input, the images were resized to 224×224x3. After the image resizing, the datasets were divided into two parts for training and testing. For the division, we incorporated two approaches. In the first one, randomly selected 70% and 30% datasets have been used for training and testing. In the other approach, i.e., leave-one-subject-out, the dataset created by the k-1 signer has been used for training and the remaining one for testing. During the testing phase, the procedure was simply repeated once for each signer. The average validation accuracy is taken into account following the k-th round. This kind of performance evaluation offers a more accurate judgement of model ability. Since some of the selected datasets do not have sufficient gestures, data augmentation was employed to prevent the problem of over-fitting. Data augmentation increases gesture images via different signal processing operations, as discussed below, for training.

- Rotation: The training dataset's images are randomly rotated up to five degrees.
- Translation: The images are randomly translated either vertically or horizontally. Their coordinates are changed throughout this operation.
- Shear: In this method, the vertical range of the original image pixels is linearly increased with the horizontal distance from a vertical line or decreased with the opposite. In our experiment, a range of 0.2 has been chosen.

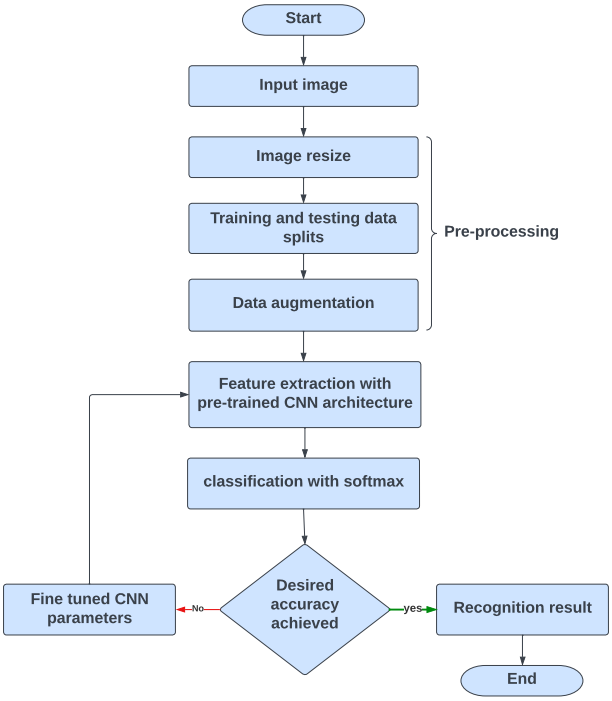

Fig. 1 The data flow diagram depicting the working model

- Zooming: In this case, randomly zoomed the images from the dataset. The zooming operation's range is assumed to be 0.9. The size of the training datasets increased sufficiently after data augmentation to overcome the overfitting issue.

4.2 Feature extraction using mIV3Net

Generally, the recognition performance of a neural network enhances with increased depth, but with the cost of high computational and time requirements. Hence, transfer learning has been developed to reduce the cost of training. Transfer learning entails transferring the model parameters from a trained network to any other model to improve the training efficiency. By sharing the parameters of the trained model using transfer learning, the new network's performance improves, rather than beginning from scratch. In the case of training from scratch, the amount of data required is very high. In contrast transfer learning approach reduces the data required during training. The number of hand gesture images in some selected datasets is not sufficient to train the neural network model from scratch. Therefore, transfer learning has been employed. Based on the advantages listed in the literature, empirically, we have chosen the inception V3 network [44] as the transfer learning, trained on an image net dataset (including more than one million copies having 1000 categories of image data). Without transfer learning, if we train the inception V3 network from scratch using a low-configured computer, it will take at least a few days to train it. A customized version of inception V3 has been used for feature extraction. The inception modules that are utilized to replace the convolution layers are one of the innovative features of inception V3. The

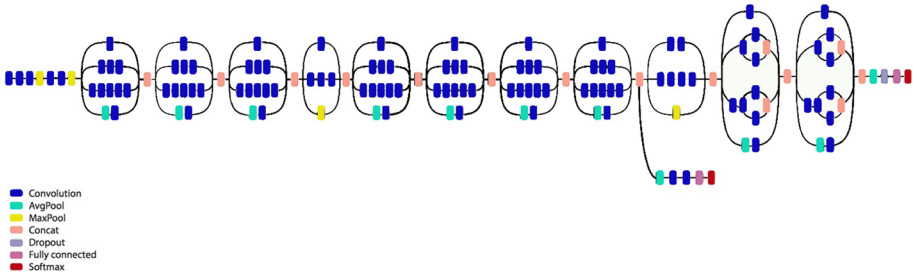

Fig. 2 The architecture of inception V3

inception module uses several conventional convolutional layers to extract features, and the result is a concatenation of the extracted feature. The inceptionV3 module contrast with a traditional convolution layer due to feature extraction from varied kernel size. Consequently, the extracted feature is not constrained to a fixed-scale local region. When the inception module is used to extract gesture characteristics, different kernel size helps the model to generalize for the various size. Inception V3 architecture is shown in Fig. 2. *Inception V3 architecture is improved and named mIV3Net: modified inception V3 network to reduce the computational resource requirement*, as shown in Fig. 3. Here, we empirically selected the first eight concatenation modules of the inception V3, excluding the other modules. The reason for this modification is that the recent CNNs have too much depth [23], hence needing large memory and computational resources. Additionally, these models reduce the HGR's effectiveness by failing to encode the proper and necessary features from datasets. These observations led us to develop mIV3Net, which identifies the most crucial features for accurately classifying gestures. mIV3Net's less clumsy design makes possible to deploy it in low resource environment. Additionally, the suggested mIV3Net does not require segmentation of only the palm part of gestures, which simplifies the recognition process. After selecting the first eight concatenation modules of the inception V3, it is extended by adding zero padding, a convolution layer with 512 filters. Zero-padding is a general method for preventing information loss at the boundaries and controlling the decrease of sizes when using filters greater than 1×1. *The modification mainly includes: the selection of appropriate layers;*

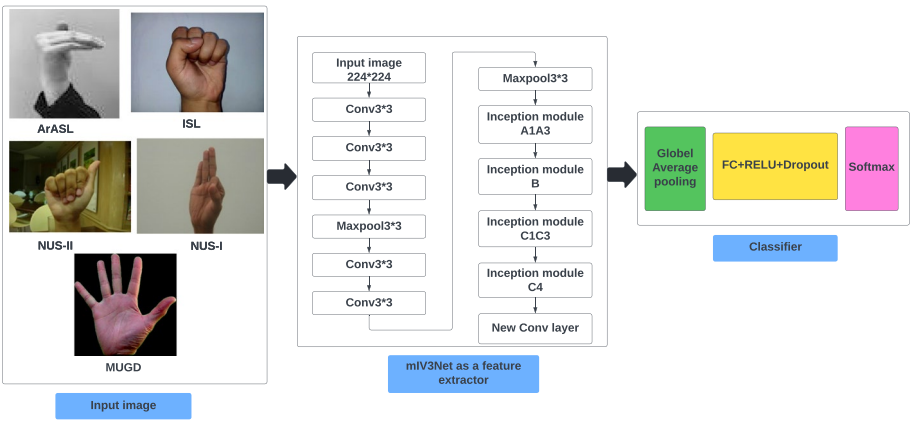

Fig. 3 The Proposed algorithms for hand gesture recognition

Fig. 4 The sample images in the datasets (**a**) NUS-II (**b**) NUS-I (**c**) ArSL (**d**) ISL (**e**) MUGD

dropout value; addition of a new layer in order to extract more detailed features, etc. Extensive experiments are administered to attain the most salient layer features for recognition. Convolutional layers are useful for the extraction of features from images because they use weight sharing to address spatial redundancy. Redundancy decreases, and features get more specialized and informative as we move further into the network. This is mostly caused by the compression of information using subsampling layers and repeatedly cascading convolutions. The newly added convolutional layer will learn more information related to the selected hand gesture datasets.

4.3 Classifier

The feature extractor designed in the earlier sub-section is extended using a newly added classifier. The new classifier consists of global average pooling, dropout, ReLu activation, fully connected layers, and SoftMax classifier. The global average pooling (GAP) maps features into a more robust form for a better understanding of patterns. In this paper, the fattening layer has been replaced with the GAP layer for better accuracy. It also reduces the problem of

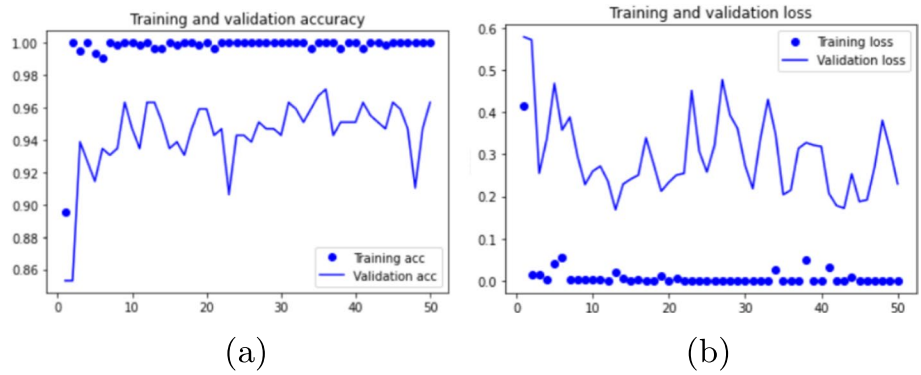

Fig. 5 **a** Accuracy vs epoch plot for MUGD dataset (x-coordinate is epoch and y-coordinate is accuracy). **b** Loss vs epoch plot for MUGD dataset (x-coordinate is epoch and y-coordinate is loss)

overfitting. A dropout is applied before the fully connected layers as means of regularization. On top of dropout, a dense layer with a SoftMax classifier is used for classifying gestures into the corresponding class. Through experimental analysis, a suitable dropout rate and the number of convolution filters have been selected to obtain better gesture recognition accuracy.

4.4 Fine-tuning

We have fine-tuned the architecture and layers of an earlier trained model. In this approach, the earlier combined architecture is empirically modified as follows. The first four concatenation modules are frozen, and the remaining modules are retrained for better feature extraction. This fine-tuning is introduced to offer attention to important features. As a result, better abstract knowledge has been used for gesture recognition.

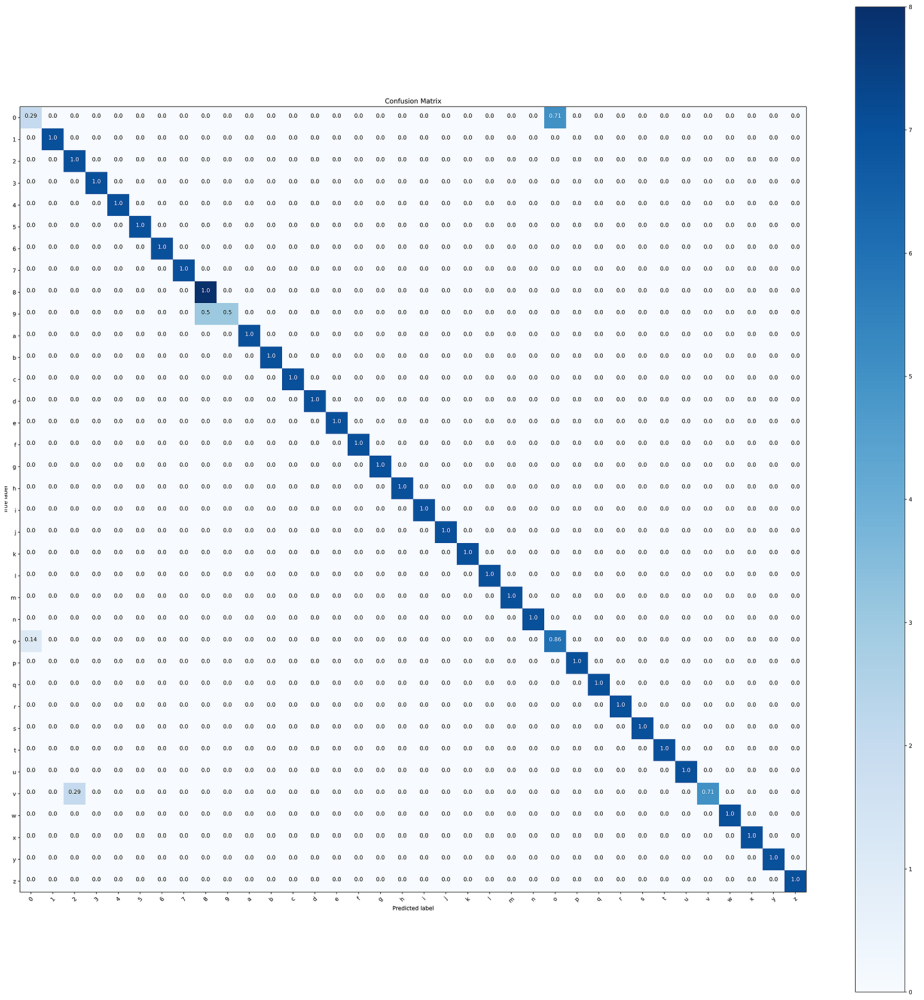

Fig. 6 Confusion matrix for MUGD dataset

5 Experimental results and discussions

In this section, we have first presented the implementation and training details of mIV3Net. Next, the performance metrics and five publicly available datasets are described. Further, the quantitative and qualitative results are presented to demonstrate the efficacy of mIV3Net. Then, we discuss the computational complexity of the proposed system. Finally, the importance of mIV3Net is drawn in the ablation study.

5.1 Implementation and training details

The experiments have been conducted using Keras. mIV3Net is trained using the following hyperparameters, batch size: 16, dropout rate: 0.4, cost function: cross-entropy, optimizer: RMSprop, and learning rate: 0.0004. All HGR algorithms are implemented using

Table 1 The performance Analysis of MUGD dataset

Class	Precision	Recall	F1-score
a	1.00	1.00	1.00
b	1.00	1.00	1.00
c	1.00	1.00	1.00
d	1.00	1.00	1.00
e	1.00	1.00	1.00
eight	1.00	1.00	1.00
f	1.00	1.00	0.92
five	1.00	1.00	1.00
four	1.00	1.00	1.00
g	1.00	1.00	0.92
h	1.00	1.00	1.00
i	1.00	1.00	1.00
j	1.00	1.00	1.00
k	1.00	1.00	1.00
l	1.00	1.00	1.00
m	1.00	1.00	1.00
n	1.00	1.00	1.00
nine	1.00	1.00	1.00
o	0.60	0.86	0.71
one	1.00	1.00	1.00
p	1.00	1.00	1.00
q	1.00	1.00	1.00
r	1.00	1.00	1.00
s	1.00	1.00	1.00
seven	1.00	0.71	0.83
six	1.00	1.00	1.00
t	1.00	1.00	1.00
three	1.00	1.00	1.00
two	0.78	1.00	0.88
u	1.00	1.00	1.00
v	0.71	0.71	0.71

the online KAGGLE GPU kernel with Tesla P100 and 16 GB VRAM. The Laptop configuration is 11th Gen Intel(R) Core(TM) i7 and 16 GB RAM. The datasets are augmented to promote generalization and prevent over-fitting during training. We empirically selected the first eight concatenation modules of the inception V3. Next, it is extended by adding zero padding, and convolution layer with 512 filters, which yields a feature extractor. Finally, the feature extractor is expanded using a newly added densely connected classifier named mIV3Net. We train the mIV3Net using the selected datasets. The weight from ImageNet has been used to initialize the feature extractor's first eight concatenation modules. We fine-tuned mIV3Net by freezing the first four concatenation modules. Initial layer freezing prevents them from changing their weights while training. The initial few layers were frozen for two reasons. Firstly, our datasets differ significantly from those used by ImageNet. Secondly, slightly more depth layers of the feature extractor contain more specialized features. The earlier layers have more generic and reusable features. Finally, we trained this fine-tuned model and achieved better results, as discussed in subsection 5.9.

5.2 Evaluation metric

The efficacy of the HGR system has been evaluated using accuracy, precision, recall, and F1-score. Accuracy measures correctly classified classes, whereas F1-score dealings incorrectly identified classes. The weighted harmonic means of precision and recall have been considered for calculating the F1-score. The accuracy and F1-score of mIV3Net have been evaluated for similar and imbalanced class distributions. The performance metrics have been mathematically expressed as follows.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

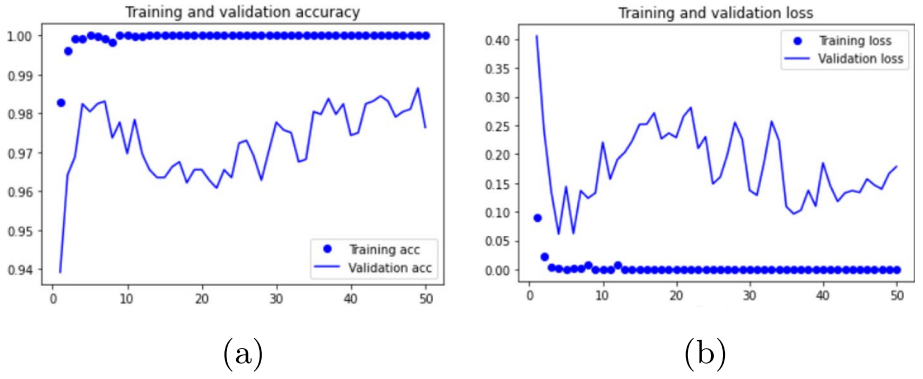

Fig. 7 **a** Accuracy vs epoch plot for ISL dataset (x-coordinate is epoch and y-coordinate is accuracy). **b** Loss vs epoch plot for ISL dataset (x-coordinate is epoch and y-coordinate is loss)

$$\text{F1-score} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

where TP, FP, TN, and FN represent true positive, false positive, true negative, and false negative, respectively.

5.3 Datasets

MUGD [4], ISL [3], ArSL [20], NUS-I [36], and NUS-II [36] datasets have been used to test mIV3Net and compared methods. These datasets have been created from images taken with varying illumination and cluttered backgrounds. Figure 4 depicts some sample images from the datasets.

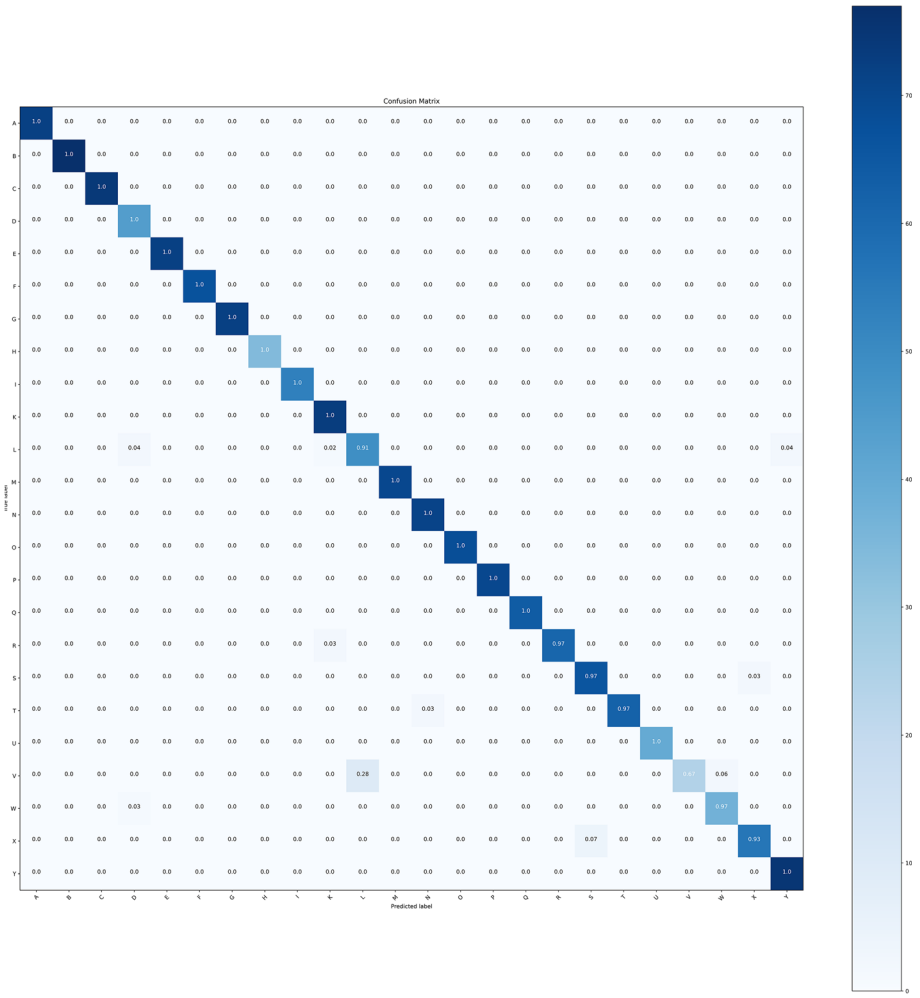

Fig. 8 Confusion matrix for ISL dataset

Table 2 The performance Analysis of ISL dataset

Class	Precision	Recall	F1-score
A	0.95	1.00	O.97
B	0.97	1.00	0.99
C	1.00	0.99	0.99
D	0.96	0.98	0.97
E	1.00	1.00	1.00
F	1.00	0.97	0.98
G	1.00	1.00	1.00
H	1.00	0.97	0.99
I	1.00	0.94	0.97
K	1.00	1.00	1.00
L	0.79	1.00	0.89
M	0.88	0.96	0.92
N	1.00	0.87	0.93
O	1.00	1.00	1.00
P	1.00	1.00	1.00
Q	1.00	1.00	1.00
R	1.00	0.92	0.96
S	0.97	1.00	0.99
T	1.00	1.00	1.00
U	1.00	1.00	1.00
V	1.00	0.72	0.84
W	0.95	1.00	0.97
X	1.00	1.00	1.00
Y	1.00	1.00	1.00

Massey University Gesture Dataset (MUGD) In this dataset, 2524 images from ASL gestures are included in the MUGD, which consists of 36 distinct alphabets, i.e., classes from a to z and integers from 0 to 9. The images were taken with a consistent black background from five participants in five distinct directions, including left, right, bottom, and top.

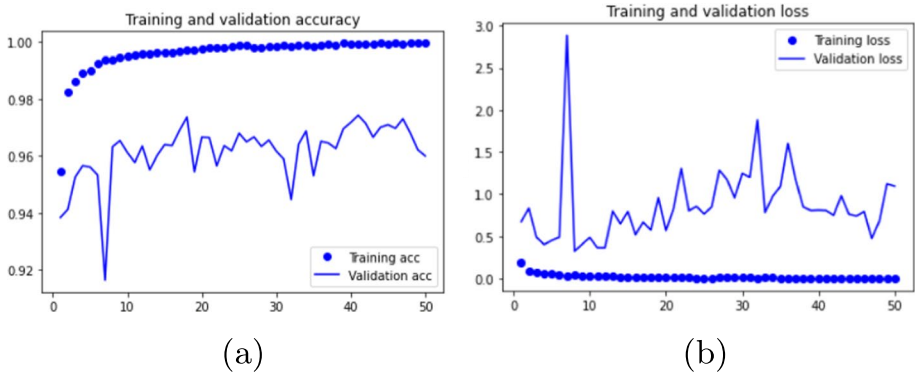

Fig. 9 **a** Accuracy vs epoch plot for ArSL dataset (x-coordinate is epoch and y-coordinate is accuracy). **b** Loss vs epoch plot for ArSL dataset (x-coordinate is epoch and y-coordinate is loss)

ISL dataset As no standardized dataset is available for ISL, gestures are collected from the online provided in [3]. Except for J and Z, this dataset includes 200 images for each ISL alphabet set and the total consists of 4962 images. Figure 4 depicts the gestures for this dataset.

Arabic dataset (ArSL) This dataset is publicly available in [20], which contains 32 ArSL classes and 54,094 grayscale images with various lighting and backgrounds. Figure 4 shows the ArSL samples.

NUS-I dataset Ten gesture classes, with 24 example images each, can be found in this dataset. Here, the hand gestures have recorded by altering the subject's position and size of the hand.

NUS-II dataset This dataset has ten classes using alphabets starting from a to j. The postures are carried out by many individuals with various hand sizes and shapes with a complex

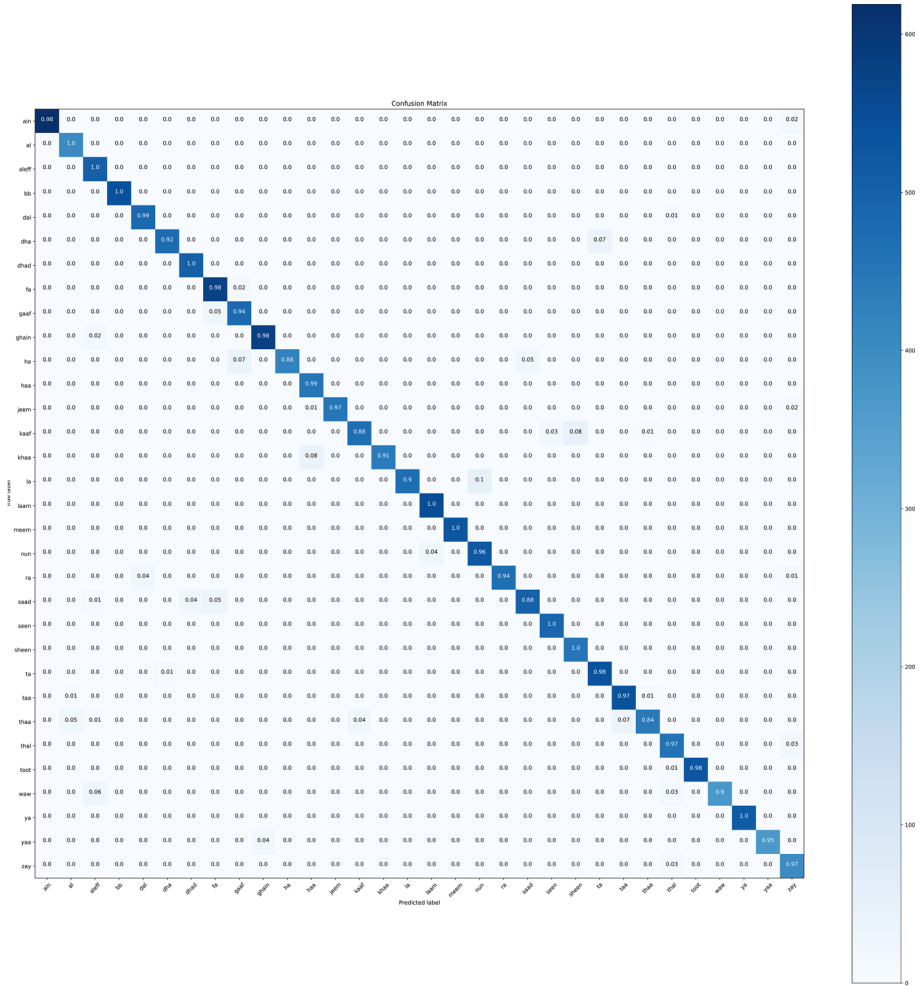

Fig. 10 Confusion matrix for ArSL dataset

background to incorporate natural variances. The dataset is challenging because of the gestures performed by forty people of various societies, genders, and ages (in years) from 22 to 56.

5.4 Quantitative analysis

mIV3Net has been trained by the aforementioned training configuration for five datasets. Two cross-validation techniques have been adopted to test the efficacy of the proposed method. In the first method, a random 70:30 split was done between the training and testing portions of the datasets. In the next one, the leave-one-subject-out approach has been used to check the generality of the HGR system across various datasets. The performance of the proposed HGR system considering each gesture from various datasets is presented in subsections 5.4.1 to 5.4.5. Two separate comparative analyses have been presented in subsections 5.5 and 5.6.

Table 3 The performance analysis of ArSL dataset

Class	Precision	Recall	F1-score
ain	1.00	1.00	1.00
al	0.98	1.00	0.99
aleff	0.94	1.00	0.97
bb	1.00	0.96	0.98
dal	0.98	1.00	0.99
dha	0.99	0.92	0.95
dhad	1.00	1.00	1.00
fa	o.88	1.00	0.94
gaaf	0.81	0.86	0.84
ghain	0.97	0.96	0.97
ha	0.99	0.88	0.93
haa	0.99	0.94	0.97
jeem	0.99	1.00	0.99
kaaf	1.00	0.78	0.87
khaa	0.94	0.99	0.97
la	1.00	0.93	0.96
laam	0.99	1.00	0.99
meem	0.97	1.00	0.99
nun	0.96	0.99	0.97
ra	1.00	0.97	0.98
saad	1.00	0.93	0.96
seen	0.89	0.90	0.89
sheen	0.80	1.00	0.89
ta	0.91	0.99	0.95
taa	0.95	0.98	0.97
thaa	0.99	0.95	0.97
thal	1.00	0.96	0.98
toot	0.99	0.98	0.99
waw	0.97	0.97	0.97
ya	0.98	0.97	0.98
yaa	1.00	0.93	0.96
zay	0.93	1.00	0.96

5.4.1 Results for MUGD dataset

A 70:30 split have been done between the training and testing portions of the MUGD dataset. Figure 5 depicts the accuracy curve for mIV3Net over the varying number of epochs. The validation data's confusion matrix is depicted in Fig. 6. Precision, Recall, and F1-score of gestures are presented in Table 1. Figure 6 shows that most gesture classes have been correctly classified using the proposed approach. Four gesture classes have been misclassified with the accuracy of 0.86, 0.71, 0.71, and 0.43. The accuracy of class zero is lowest due to interclass similarity with class 0 (letter). The overall accuracy of mIV3Net without fine-tuning is 90.61%, which has been improved to 97.14% due to fine-tuning. Hence, mIV3Net is performing well for the MUGD dataset due to fine-tuning.

5.4.2 Results for ISL dataset

The training and testing components of the ISL dataset were divided at 70:30. The accuracy curve for mIV3Net over various epoch counts is shown in Fig. 7. Figure 8 shows the confusion matrix for the validation data. Table 2 displays the precision, recall, and F1-score of gestures. Figure 8 shows that most gesture classes have been correctly classified using the proposed method. There are only two gesture classes, i.e., "R" and "V", which have recognition accuracy of 87% and 57%. It can be observed that class "V" has been misclassified to class "W" due to similarity in gesture performance. The overall accuracy of mIV3Net is 98.6% which has been improved to 99.3% with the help of fine-tuning. As a result of fine-tuning, mIV3Net is functioning well for the ISL dataset.

5.4.3 Results for ArSL dataset

The training and testing data of the ArSL dataset have segregated 70:30. The accuracy curve for mIV3Net over various epoch counts is shown in Fig. 9. Figure 10 shows the confusion matrix for the validation data. Table 3 displays the precision, recall, and F1-score of gestures. Figure 10 shows that accuracies for 22 gesture classes, which are more than 95%, whereas, for seven gesture classes, the accuracies are between 92–95%. The remaining three gesture classes have accuracies of 88%, 86%, and 78%. It is noteworthy that just three classes are more

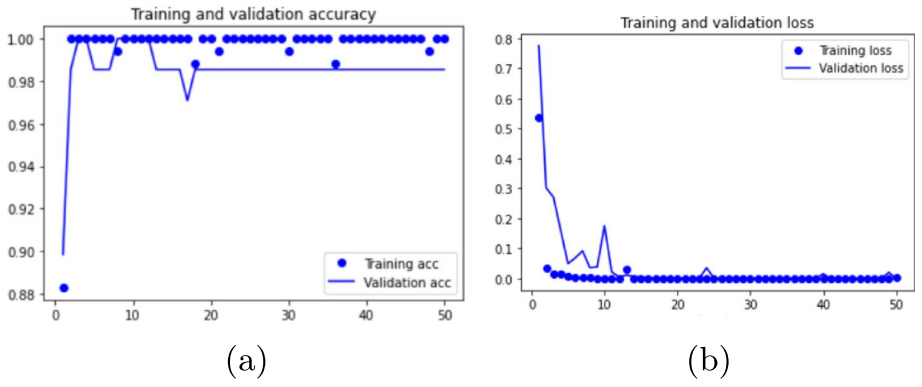

Fig. 11 **a** Accuracy vs epoch plot for NUS-I dataset (x-coordinate is epoch and y-coordinate is accuracy). **b** Loss vs epoch plot for NUS-I dataset (x-coordinate is epoch and y-coordinate is loss)

perplexed, and the remaining classes have comparable accuracy, which may be due to fine-tuning of mIV3Net. Thus, it can be inferred that mIV3Net has better classification capability.

5.4.4 Results for NUS-I dataset

The NUS-I dataset was split between training and testing halves at 70:30. Figure 11 displays the accuracy curve for mIV3Net over various epoch counts. The confusion matrix for the validation data is displayed in Fig. 12. The precision, recall, and F1-score of gestures are shown in Table 4. It can be seen from Fig. 12 that the recognition accuracy is 99%, except for one gesture class, which has an accuracy of 86%. A better accuracy may occur due to the better feature extraction capability of mIV3Net. We have noted that the validation accuracy was 91.3% without fine-tuning, which has improved to 99% due to fine-tuning.

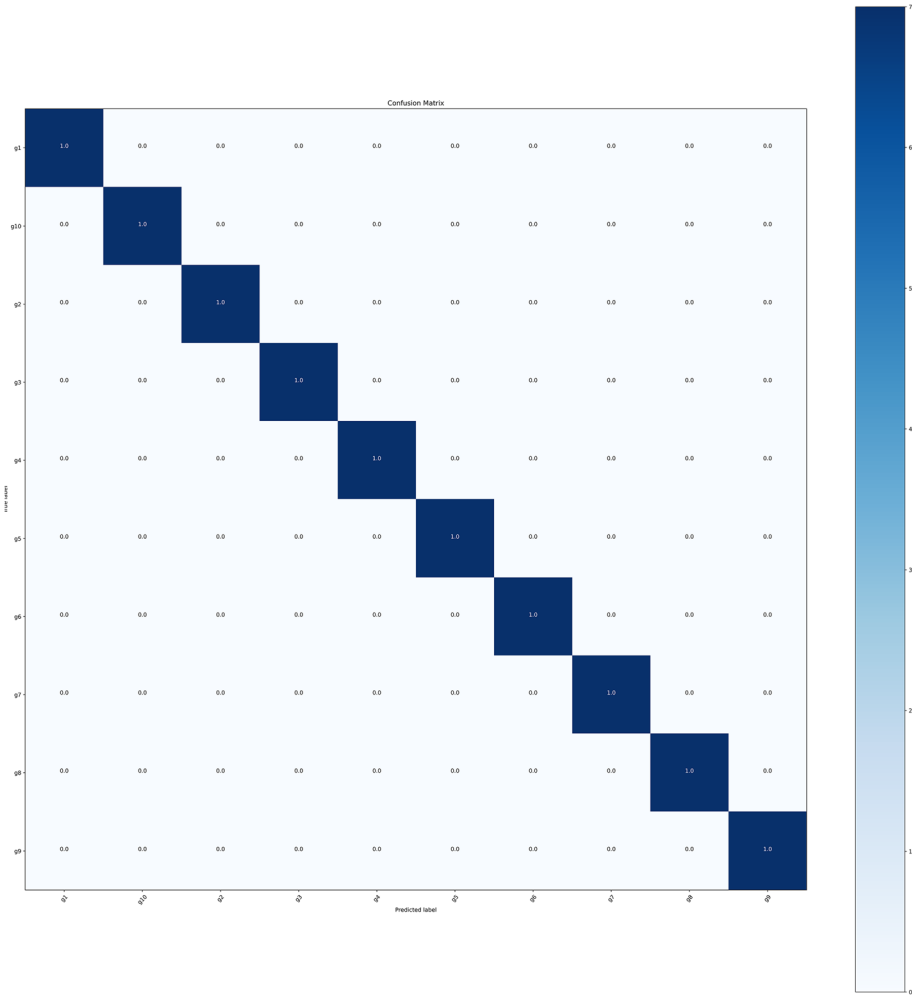

Fig. 12 Confusion matrix for NUS-I dataset

Table 4 The performance analysis of NUS-I dataset

Class	Precision	Recall	F1-score
g1	1.00	1.00	1.00
g2	1.00	1.00	1.00
g3	1.00	0.99	1.00
g4	1.00	0.98	1.00
g5	1.00	1.00	1.00
g6	1.00	0.97	1.00
g7	1.00	0.86	0.92
g8	1.00	1.00	1.00
g9	1.00	1.00	1.00
g10	0.86	1.00	0.92

5.4.5 Results for NUS-II dataset

The mIV3Net has been trained using the above-mentioned training setup and achieved an average accuracy of 99.8% on the validation dataset. The NUS-II dataset has been split for training and testing in a ratio of 7:3. An accuracy curve during training and testing over the number of epochs for mIV3Net is shown in Fig. 13. The confusion matrix for the validation data, as shown in Fig. 14. The precision, recall, and F1-score of each gesture is shown in Table 5. It can be observed from Fig. 14 that eight gesture classes have been correctly recognized, while two gesture classes, i.e., "a" and "c" have an accuracy of 98%. It can be mentioned that the overall recognition accuracy is 99.8% using fine-tuning of mIV3Net, whereas it was 90.50% without fine-tuning.

5.5 Comparative analysis of the proposed method with related recent methods using random split cross-validation

Tables 6 and 7 display the accuracy of mIV3Net and other compared techniques for the five datasets using random split. When compared to other approaches, it can be seen from Table 6 that mIV3Net achieves a greater accuracy rate. Notably, the

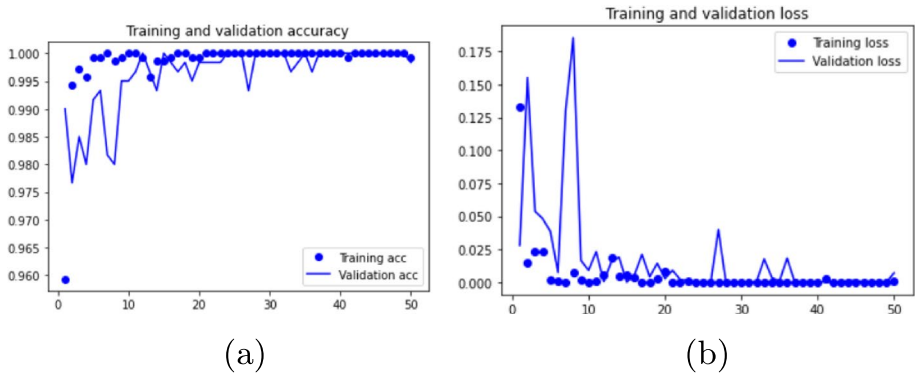

Fig. 13 **a** Accuracy vs epoch plot for NUS-II dataset (x-coordinate is epoch and y-coordinate is accuracy). **b** Loss vs epoch plot for NUS-II dataset (x-coordinate is epoch and y-coordinate is loss)

enhancements in accuracy values of mIV3Net are 12.58%, 19.2%, 20.14%, and 36.31% over HyFiNet [5], DenseNet-121 [13], ResNet-50 [11], and MobileNetV2 [42], respectively, on MUGD dataset. In another dataset, i.e., NUS-I, the proposed mIV3Net has better performance than HyFiNet [5], ResNet-50 [11], MobileNetV2 [42], and DenseNet-121 [13] in terms of accuracy by 0.56%, 7.82%, 12.28%, and 33.37%, respectively. For the complex background dataset, i.e., NUS-II, the proposed mIV3Net attains enhancement in the accuracy of 2.02%, 2.7%, 13.37%, and 14.2% over HyFiNet [5], ResNet-50 [11], DenseNet-121 [13], and MobileNetV2 [42], respectively. It can be observed that HyFiNet [5] is a second better method due to the inclusion of an attention block of hybrid features. But, gesture recognition accuracy is slightly lower than mIV3Net. Table 7 shows that for the ISL dataset, mIV3Net improvement rates are 1.84%, 5.8%, 8.4%, and 11.75% over E-WOA-Deep CNN [19], Multilevel HOG [17], mRMR- PSO [3], and TOPSIS [17], respectively. In another dataset, i.e., ArSL,

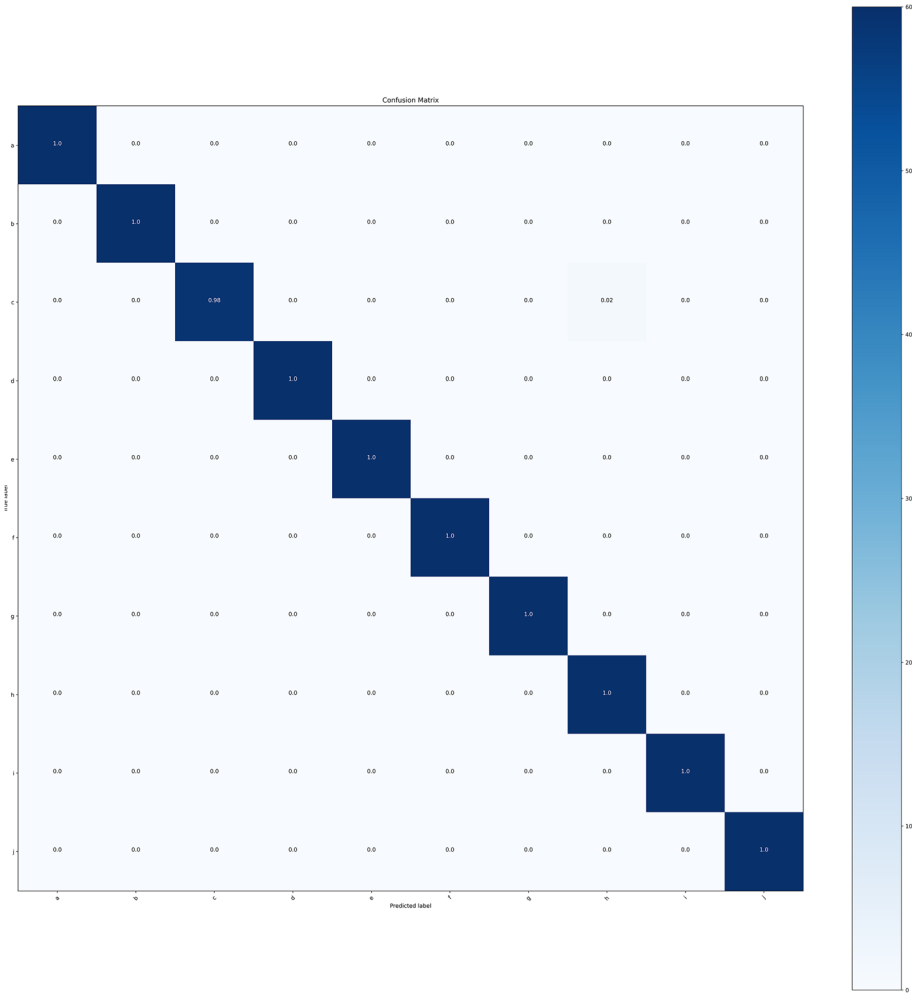

Fig. 14 Confusion matrix for NUS-II dataset

Table 5 The performance analysis of NUS II dataset

Class	Precision	Recall	F1-score
a	1.00	0.98	0.99
b	1.00	1.00	1.00
c	1.00	0.98	0.99
d	1.00	1.00	1.00
e	1.00	1.00	1.00
f	0.98	1.00	0.99
g	1.00	1.00	1.00
h	o.98	1.00	0.99
i	1.00	1.00	1.00
j	1.00	1.00	1.00

mIV3Net outperforms SIFT-LDA [46], mRMR-PSO [3], CNN [18], and Aly et al. [1] in terms of accuracy by 2.73%, 5.7%, 7.4%, and 10.55%, respectively. It can be observed that mIV3Net has a competitive performance with the SIFT-LDA [46] and outperform other methods. The SIFT-LDA paper has considered a small dataset, but the whole dataset has been used in our method. mRMR-PSO [3] is the second better method due to its better feature selection approach. In CNN [18], low accuracy is achieved due to the shallow architecture compared to the other methods. We want to mention that the recognition accuracy is 92.54%, which has sufficiently improved to 97.4% due to the proposed fine-tuning approach.

5.6 Comparative analysis of the proposed method with related recent methods using leave-one-subject-out cross-validation

To show the efficacy of mIV3Net on unseen data, leave-one-subject-out cross-validation has been employed, which provides more generalization. The comparison of mIV3Net with the state-of-the-art techniques via leave-one-subject-out cross-validation is shown in Tables 8 and 9. It can be observed that mIV3Net has achieved enhancement in accuracy values on various datasets, i.e., 9.50% to 14.66% on MUGD, 25.24% to 29.20% on NUS-I, 2.40% to 30.69% on NUS-II, 3.06% to 12.95% on ISL, and 3.27% to 11.54% on ArSL. Tables 8 and 9 show that gesture recognition accuracy values of all considered networks are lower in leave-out-subject-out cross-validation, as compared to random split. The reduction may be due to unseen data, moreover, the proposed approach has better performance than the compared ones, and shows the generalization capability.

Table 6 The performance comparision of the proposed method with the state-of-art approaches on MUGD, NUS-I, NUS-II datasets in random split

Method used	Datasets		
	MUGD	NUS-I	NUS-II
DenseNet-121 [13]	77.94	65.63	86.43
HyFiNet [5]	84.56	98.44	97.78
MobileNetV2 [42]	60.83	86.72	85.60
ResNet-50 [11]	77	91.18	97.10
mIV3Net	97.14	99	99.8

Table 7 The performance comparision of the proposed method with the state-of-art approaches on ISL, ArSL datasets in randam split

Datasets	Method used	Accuracy (%)
ISL	TOPSIS [17]	86.85
	Multilevel HOG [17]	92.8
	mRMR- PSO [3]	90.2
	E-WOA-Deep CNN [19]	96.76
	mIV3Net	99.3
ArSL	Aly et al. [1]	86.85
	SIFT-LDA [46]	94.67
	CNN [18]	90
	mRMR-PSO [3]	91.7
	mIV3Net	97.4

5.7 Qualitative analysis

Figure 15 displays the response of fine-tuned mIV3Net and the current networks on the datasets MUGD, ISL, ArSL, NUS-I, and NUS-II. In contrast to current HGR techniques, Fig. 15 demonstrates that fine-tuned mIV3Net can represent better salient features, hence achieving better accuracy. The suggested fine-tuned mIV3Net, which results in more accurate hand gesture identification, is shown to preserve the most prevalent elements necessary for differentiating hand motions, as shown in Fig. 16. As a result, the class activation map shows that mIV3Net performs better than the current cutting-edge HGR methods.

5.8 Computational load

The computational load of the mIV3Net is compared with Inception V3 architecture, as shown in Table 10. It can be observed that the number of trainable parameters has been drastically reduced, i.e., 5.9 M, as compared to Inception V3, i.e., 23.8 M. The requirement of memory for storage is also less in the proposed method, i.e., 133.64 MB, whereas 179.3 MB for Inception V3. Besides, the training time requirement is around 63% less as compared to Inception V3. Based on the experimental results and computational load requirement, it can be inferred that the proposed mIV3Net with fine-tuning provides generalized solutions for HGR. Also, considering the inference time, it may be used for real-time applications.

Table 8 The performance comparision of the proposed method with the state-of-art approaches on MUGD, NUS-I, NUS-II datasets in leave-one-subject-out

Method used	Datasets		
	MUGD	NUS-I	NUS-II
DenseNet-121 [13]	74.72	58.30	57.31
HyFiNet [5]	77.44	62.26	61.49
MobileNetV2 [42]	72.28	49.89	85.60
ResNet-50 [11]	74.42	58.49	58.11
mIV3Net	86.94	87.50	88

Table 9 The performance comparision of the proposed method with the state-of-art approaches on ISL, ArSL datasets in leave-one-subject-out

Datasets	Method used	Accuracy (%)
ISL	TOPSIS [17]	79.73
	Multilevel HOG [17]	85.66
	mRMR- PSO [3]	83.03
	E-WOA-Deep CNN [19]	89.62
	mIV3Net	89.68
ArSL	Aly et al. [1]	79.71
	SIFT-LDA [46]	87.53
	CNN [18]	82.86
	mRMR-PSO [3]	84.58
	mIV3Net	87.8

5.9 Ablation study

An ablation study has been conducted to assess the effect of mIV3Net and to validate the efficacy of fine-tuning. We have performed two experiments to justify the above claim. The first

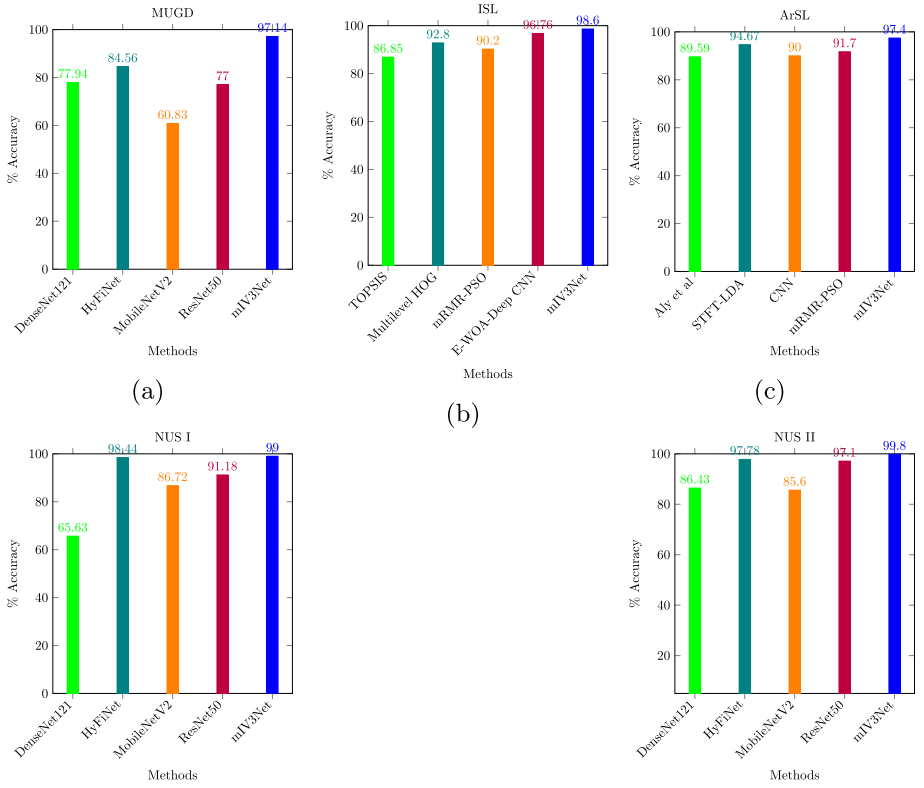

Fig. 15 The graphical representation of accuracy on the datasets (**a**) MUGD, (**b**) ISL, (**c**) ArSL, (**d**) NUSI, and (**e**) NUS-II. (x-coordinate is methods, and y-coordinate is accuracy, numbers in the panels show the accuracy of the methods)

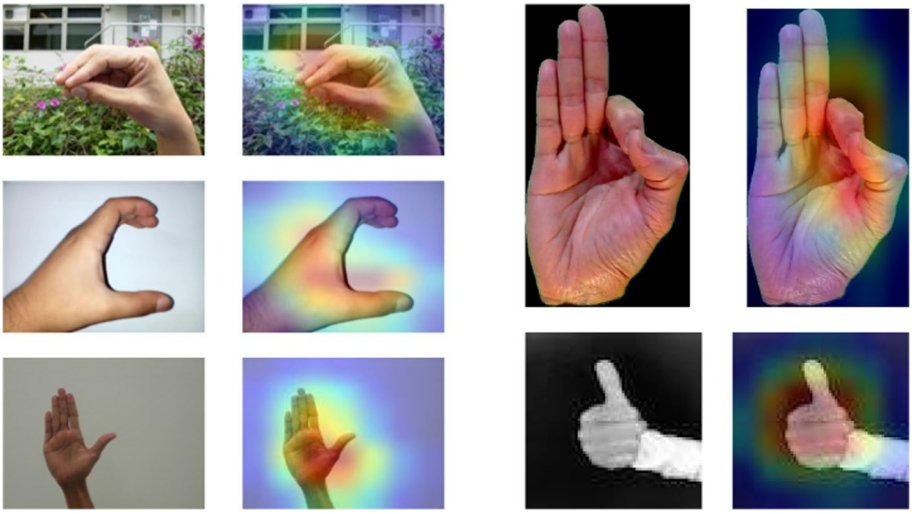

Fig. 16 The class activation map of mIV3Net on MUGD, NUS-I, NUS-II, ISL, and ArSL datasets

experiment assesses the effect of selecting an appropriate number of concatenation modules from Inception V3. The Inception V3 network is modified by empirically selecting the first eight concatenation modules and excluding the remaining ones. The experimental results for various ranges of concatenation modules for five selected datasets have been presented in Table 11. It is evident that 1–8 concatenation modules attain an increase in accuracy values over 1–7 and 1–9, i.e., 11.24% and 4.1% for MUGD, 2.25% and 2.23% for ArSL, 1.91% and 2.32% for ISL, 4.16% and 2.73% for NUS-I, and 3.67% and 7.33% for NUS-II, respectively. The top concatenation modules have more abstract knowledge of the ImageNet dataset. The initial weights on the top concatenation modules may not be helpful for the hand gesture dataset. The initial eight concatenation modules capture the salient, refined edge information, discriminable semantic structure, and fine features of hand signs. Deep models' initial convolution layers are more likely to extract finer details than deeper layers. As a result, the feature quality gradually deteriorates at the deeper layer, leading to a gradient saturation problem. This issue has been fixed by adding a new convolution layer block that adds low-level features discovered from the chosen hand gesture dataset to the top-layer features. The effectiveness of the proposed fine-tuned mIV3Net is assessed in the second experiment. We have fine-tuned the mIV3Net by empirically retraining the network from the fourth concatenation module on selected hand gesture datasets, which capture more prominent features of the hand gesture dataset. The experimental results by varying the number of concatenation modules for retraining are shown in Table 11. It can be observed that retraining from the fourth achieves better than the others. Due to fine-tuning, the recognition accuracy values have been improved from

Table 10 Comparision of computational load of mIV3Net with Inception V3. Here the letters M, MB, S stands for millions, megabytes, seconds respectively

Method used	Memory	#Parameters	Training time
Inception V3	179.3 MB	23.85 M	697.954 S
mIV3Net	133.64 MB	16.5 M	256.890 S

Table 11 The performance comparision of the proposed method "Without fine-tunning" and "With fine-tunning"

Datasets	Without fine-tunning			With fine-tunning		
	(1–7)	(1–8)	(1–9)	(from 3)	(from 4)	(from 5)
MUGD	79.37	90.61	86.51	95.63	97.14	93.89
ArSL	90.29	92.54	90.31	96.48	97.4	98.32
ISL	96.69	98.6	96.28	99	99.3	98.85
NUS-I	87.14	91.3	98.57	98.23	99	97.59
NUS-II	86.83	90.50	83.17	99.39	99.8	98.4

90.61% to 97.14% on the MUGD dataset, 98.62% to 99.3% on the ISL dataset, 92.54% to 97.4% on the ArSL dataset, and 91.32% to 99% on the NUS-I dataset, and 90.5% to 99.8% on the NUS-II dataset. Table 11 shows that the proposed approach (with fine-tuned) provides a better classification of gestures due to fine-tuned. Nevertheless, the accuracy of the suggested method is also better than some existing techniques, even without being fine-tuned.

6 Conclusions and future works

mIV3Net: Modified inceptionV3 network, a lightweight, portable CNN-based network, is suggested in the study for effective hand gesture identification. mIV3Net is simpler to implement in a limited-resource environment due to its simple architectural design. mIV3Net has been fine-tuned and generalized using five publicly available datasets. The fine-tuned mIV3Net provides better salient features, hence achieving better accuracy. The suggested fine-tuned mIV3Net, which results in more accurate hand gesture identification, is shown to preserve the most prevalent elements necessary for differentiating hand gestures. Extensive experimentation has been conducted on five datasets: MUGD, ISL, ArSL, NUS-I, and NUS-II of distinct languages under various conditions like complex background, uniform background, and varying cell size, to validate the mIV3Net. The experimental results demonstrate that in terms of classification accuracy, mIV3Net outperforms pre-trained models. The accuracy values of the proposed system on five datasets in the above order are 97.14%, 99.3%, 97.4%, 99%, and 99.8%, which are enhanced by 12.58%, 2.54%, 2.73%, 0.56%, and 2.02%, respectively, than the existing methods. In future work, some more deep neural networks may be used as ensemble learning for better classification accuracy.

Authors' contributions BK was involved in experimentation, investigations, analysis, paper writing. RHL contributed to conceptualization, reviewing and editing, supervision. RKK contributed to analysis and reviewing.

Data Availability The datasets that support the findings of this study are available in: (1) MUGD: http://hdl.handle.net/10179/4514. (2) ISL: https://doi.org/10.1007/s13369-021-06456-z. (3) ArSL: https://doi.org/10.1016/j.dib.2019.103777. (4) NUS-I and II: https://doi.org/10.1007/s11263-012-0560-5.

Declarations

Conflict of 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.

References

1. Aly S, Aly W (2020) DeepArSLR: a novel signer-independent deep learning framework for isolated Arabic sign language gestures recognition. IEEE Access 8:83199–83212
2. Badi H (2016) Recent methods in vision-based hand gesture recognition. Int J Data Sci Anal 1(2):77–87
3. Bansal SR, Wadhawan S, Goel R (2022) mRMR-PSO: a hybrid feature selection technique with a multiobjective approach for sign language recognition. Arab J Sci Eng 47(8):10365–10380
4. Barczak ALC, Reyes NH, Abastillas M, Piccio A, Susnjak T (2011) A new 2D static hand gesture colour image dataset for ASL gestures. Research Letters in the Information and Mathematical Sciences 15:12–20. http://hdl.handle.net/10179/4514
5. Bhaumik G, Verma M, Govil MC, Vipparthi SK (2023) HyFiNet: hybrid feature attention network for hand gesture recognition. Multimed Tools Appl 82(4):4863–4882
6. Can C, Kaya Y, Kılıç F (2021) A deep convolutional neural network model for hand gesture recognition in 2d near-infrared images. Biomed Phys Eng Express 7(5):055005
7. Chevtchenko SF, Vale RF, Macario V, Cordeiro FR (2018) A convolutional neural network with feature fusion for real-time hand posture recognition. Appl Soft Comput 73:748–766
8. Dadashzadeh A, Targhi AT, Tahmasbi M, Mirmehdi M (2019) HGR-Net: a fusion network for hand gesture segmentation and recognition. IET Comput Vision 13(8):700–707
9. Gupta B, Shukla P, Mittal A (2016) K-nearest correlated neighbor classification for Indian sign language gesture recognition using feature fusion. In: 2016 International Conference on Computer Communication and Informatics (ICCCI). IEEE, pp 1–5
10. Hasan HS, Kareem SA (2012) Human computer interaction for vision based hand gesture recognition: a survey. In: 2012 international conference on Advanced Computer Science Applications and Technologies (ACSAT). IEEE, pp 55–60
11. He K., Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
12. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprintarXiv:1704.04861
13. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
14. Huesser C, Schubiger S, Çöltekin A (2021) Gesture interaction in virtual reality: a low-cost machine learning system and a qualitative assessment of effectiveness of selected gestures vs. gaze and controller interaction. In: Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part III 18. Springer, pp 151–160
15. Jadooki S, Mohamad D, Saba T, Almazyad AS, Rehman A (2017) Fused features mining for depth-based hand gesture recognition to classify blind human communication. Neural Comput Appl 28:3285–3294
16. Jaramillo-Yánez A, Benalcázar ME, Mena-Maldonado E (2020) Real-time hand gesture recognition using surface electromyography and machine learning: a systematic literature review. Sensors 20(9):2467
17. Joshi G, Singh S, Vig R (2020) Taguchi-TOPSIS based HOG parameter selection for complex background sign language recognition. J Vis Commun Image Represent 71:102834
18. Kamruzzaman MM (2020) Arabic sign language recognition and generating Arabic speech using convolutional neural network. Wirel Commun Mob Comput pp 1–9. https://doi.org/10.1155/2020/3685614
19. Kowdiki M, Khaparde A (2022) Adaptive hough transform with optimized deep learning followed by dynamic time warping for hand gesture recognition. Multimed Tools Appl:1–32
20. Latif G, Mohammad N, Alghazo J, AlKhalaf R, AlKhalaf R (2019) Arasl: Arabic alphabets sign language dataset. Data Brief 23:103777
21. Li X, Deng Q (2021) Chinese position segmentation based on ALBERT-BiGRU-CRF model. In: 2021 International Symposium on Computer Technology and Information Science (ISCTIS). IEEE, pp 116–120
22. Li S-Z, Yu B, Wu W, Su S-Z, Ji R-R (2015) Feature learning based on SAE–PCA network for human gesture recognition in RGBD images. Neurocomputing 151:565–573
23. Li Y, Wang X, Liu W, Feng B (2018) Deep attention network for joint hand gesture localization and recognition using static RGB-D images. Inf Sci 441:66–78
24. Li G, Zhang L, Sun Y, Kong J (2019) Towards the sEMG hand: internet of things sensors and haptic feedback application. Multimed Tools Appl 78:29765–29782
25. Lin H-I, Hsu M-H, Chen W-K (2014) Human hand gesture recognition using a convolution neural network. In: 2014 IEEE international Conference on Automation Science and Engineering (CASE). IEEE, pp 1038–1043
26. Liu P, Li X, Cui H, Li S, Yuan Y (2019) Hand gesture recognition based on single-shot multibox detector deep learning. Mob Inf Syst 2019:1–7

27. Mujahid A, Awan MJ, Yasin A, Mohammed MA, Damaševičius R, Maskeliūnas R, Abdulkareem KH (2021) Real-time hand gesture recognition based on deep learning YOLOv3 model. Appl Sci 11(9):4164
28. Nagarajan S, Subashini T (2013) Static hand gesture recognition for sign language alphabets using edge oriented histogram and multi class SVM. Int J Comput Appl 82(4):28–35
29. Neethu P, Suguna R, Sathish D (2020) An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. Soft Comput 24:15239–15248
30. Oudah M, Al-Naji A, Chahl J (2020) Hand gesture recognition based on computer vision: a review of techniques. J Imaging 6(8):73
31. Oyedotun OK, Khashman A (2017) Deep learning in vision-based static hand gesture recognition. Neural Comput Appl 28(12):3941–3951
32. Ozcan T, Basturk A (2019) Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition. Neural Comput Appl 31:8955–8970
33. Pabendon E, Nugroho H, Suheryadi A, Yunanto PE (2017) Hand gesture recognition system under complex background using spatio temporal analysis. In: 2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME). IEEE, pp 261–265
34. Pavlovic VI, Sharma R, Huang TS (1997) Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans Pattern Anal Mach Intell 19(7):677–695
35. Pinto RF, Borges CD, Almeida AM, Paula IC (2019) Static hand gesture recognition based on convolutional neural networks. J Electr Comput Eng 2019:1–12
36. Pisharady PK, Vadakkepat P, Loh AP (2013) Attention based detection and recognition of hand postures against complex backgrounds. Int J Comput Vis 101:403–419
37. Ranga V, Yadav N, Garg P (2018) American sign language fingerspelling using hybrid discrete wavelet transform-Gabor filter and convolutional neural network. J Eng Sci Technol 13(9):2655–2669
38. Rastgoo R, Kiani K, Escalera S (2021) Sign language recognition: a deep survey. Expert Syst Appl 164:113794
39. Rathi P, Kuwar Gupta R, Agarwal S, Shukla A (2020) Sign language recognition using resnet50 deep neural network architecture. In: 5th International Conference on Next Generation Computing Technologies (NGCT-2019)
40. Rautaray SS, Agrawal A (2015) Vision based hand gesture recognition for human computer interaction: a survey. Artif Intell Rev 43:1–54
41. Rubin Bose S, Sathiesh Kumar V (2021) In-situ identification and recognition of multi-hand gestures using optimized deep residual network. J Intell Fuzzy Syst 41(6):6983–6997
42. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
43. Shanthakumar VA, Peng C, Hansberger J, Cao L, Meacham S, Blakely V (2020) Design and evaluation of a hand gesture recognition approach for real-time interactions. Multimed Tools Appl 79:17707–17730
44. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
45. Tan YS, Lim KM, Lee CP (2021) Hand gesture recognition via enhanced densely connected convolutional neural network. Expert Syst Appl 175:114797
46. Tharwat A, Gaber T, Hassanien AE, Shahin MK, Refaat B (2015) Sift-based Arabic sign language recognition system. In: Afro-European conference for industrial advancement: proceedings of the first international Afro-European Conference for Industrial Advancement AECIA 2014. Springer, pp 359–370
47. Tsai T-H, Huang C-C, Zhang K-L (2020) Design of hand gesture recognition system for human-computer interaction. Multimed Tools Appl 79:5989–6007
48. Von Hardenberg C, Bérard F (2001) Bare-hand human-computer interaction. In: Proceedings of the 2001 workshop on perceptive user interfaces, pp 1–8
49. Wadhawan A, Kumar P (2020) Deep learning-based sign language recognition system for static signs. Neural Comput Appl 32:7957–7968
50. Wang C, Liu Z, Chan S-C (2014) Superpixel-based hand gesture recognition with Kinect depth camera. IEEE Trans Multimed 17(1):29–39
51. Xie B, He X, Li Y (2018) RGB-D static gesture recognition based on convolutional neural network. J Eng 2018(16):1515–1520
52. Yasen M, Jusoh S (2019) A systematic review on hand gesture recognition techniques, challenges and applications. PeerJ Comput Sci 5:218

53. Zakariah M, Alotaibi YA, Koundal D, Guo Y, Mamun EM (2022) Sign language recognition for Arabic alphabets using transfer learning technique. Comput Intell Neurosci pp 1–15. https://doi.org/10.1155/2022/4567989
54. Zhang T, Lin H, Ju Z, Yang C (2020) Hand gesture recognition in complex background based on convolutional pose machine and fuzzy Gaussian mixture models. Int J Fuzzy Syst 22:1330–1341
55. Zhang W, Wang J, Lan F (2020) Dynamic hand gesture recognition based on short-term sampling neural networks. IEEE/CAA J Autom Sin 8(1):110–120
56. Zhao J, Allison RS (2020) Comparing head gesture, hand gesture and gamepad interfaces for answering yes/no questions in virtual environments. Virtual Real 24(3):515–524
57. Zhou W, Chen K (2022) A lightweight hand gesture recognition in complex backgrounds. Displays 74:102226