Multimedia Tools and Applications (2025) 84:3137–3159
https://doi.org/10.1007/s11042-024-20585-1

# Effi-CNN: real-time vision-based system for interpretation of sign language using CNN and transfer learning

**Pranav**[1] · **Rahul Katarya**[1]

Received: 5 September 2022 / Revised: 13 May 2023 / Accepted: 26 December 2024 /
Published online: 7 January 2025


**Abstract**
The deaf and mute population has difficulty conveying their thoughts and ideas to others. Sign language is their most expressive mode of communication, but the general public is callow of sign language; therefore, the mute and deaf have difficulty communicating with others. A system that can correctly translate sign language motions to speech and vice versa in real time is required to overcome this communication barrier. Effi-CNN, a vision-based Sign Language Recognition (SLR) system employing transfer learning with EfficientNetB2, is proposed in this paper. We have also developed a system covering sign gestures to text in real time. Our approach was evaluated on eight publicly available datasets, including the Massey University gesture dataset, the ArSL2018 dataset, the MNIST-ASL dataset, and others. When comparing our results to state-of-the-art algorithms, the experimental findings showed that our proposed work is more successful than existing models. Our proposed model delivered an accuracy of 99.90 on MNIST Dataset. The results show that our Effi-CNN surpasses most currently existing solutions and can categorize many gestures with a low error rate.

**Keywords** Convolutional Neural Network (CNN) · Deep Learning (DL) · Sign Language (SL) · Transfer Learning

## 1 Introduction

Not everyone can hear or talk; to connect with society, such people have adopted unique communication processes rather than their voices. People with hearing and speech difficulties use hand gestures and accompanying motions to express their intended ideas [1]. Sign Language (SL) is a set of actions organized according to syntax, grammar, semantics, pragmatics, and morphology. For persons who are deaf or mute, SL happens to be

✉ Rahul Katarya
rahuldtu@gmail.com

Pranav
prrnnv@gmail.com

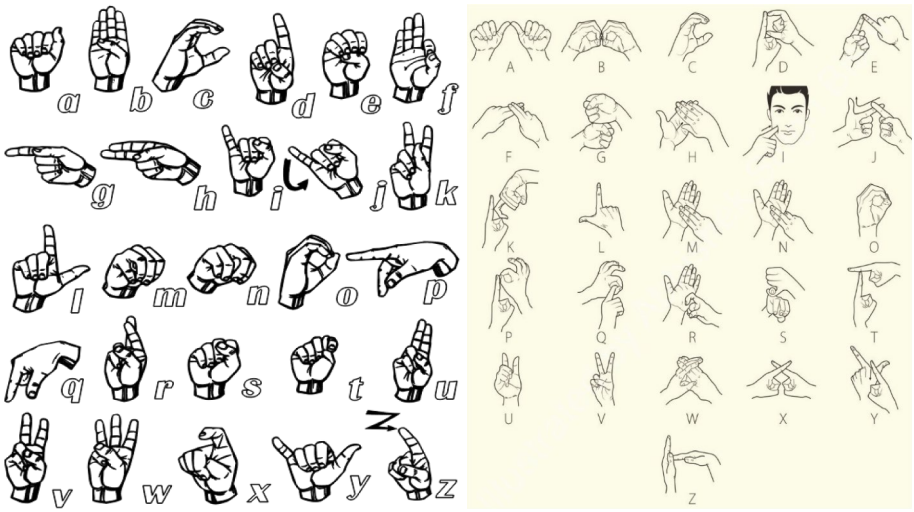
[1] Big Data Analytics and Web Intelligence Laboratory, Computer Science & Engineering Department, Delhi Technological University, New Delhi, India

the natural way of communication [2]. There is no universal Sign Language (SL), and SLs have evolved spontaneously as diverse groups of people engage with one another. Different nations and areas utilize other SLs, and practically every country has an official SL [3]; examples include Indian SL (ISL), Arabic SL (ArSL), American SL (ASL), etc. Even in countries with the same spoken language, the sign languages used are different, e.g. British SL and ASL. In Fig. 1, it can be seen that the alphabet 'Q' of ASL [4] looks similar to the alphabet 'U' of Indian SL (ISL) [5]. Just like spoken languages, SLs differ in terms of most linguistic aspects, like grammar structure [6]. One specific SL may differ from another in the essential alphabetical representation, as shown in Fig. 1, including the shape of the sign and the way the sign is performed [3, 7, 8]. In this work, datasets from three different SLs are chosen for study.

Literature prominently cites 2 types of techniques for SLR systems: vision-based techniques (VBT) and sensor-based gesture techniques (SBT); other than these, some other systems employing RADAR waves [9], speakers [10], EEG sensors[11–14], Wi-Fi modules [15–18], RF waves [19], etc. have also been used for the same.

The VBT is an artificial intelligence-based solution to sign gesture identification challenges that use cameras and image processing techniques. Many researchers have used a single camera to record indications as the default method. Edge-based active contours [20], frame subtraction [21], wavelet transform [22], and other approaches are used to segment the signer's hand from gesture images. Some researchers have employed RGB-D sensors to gather depth data of the signer's hand and body, such as Microsoft Kinect [23–26]. Another device mentioned in the literature [27, 28] that records hand motions on a 3D axis is the leap motion sensor. The detailed photos produce 3D data of the hand and body of the signer. Many critical metrics may be calculated using this data, including hand direction, wrist orientation, and joint angles.

In SBT, wearable devices consisting of sensors such as Flex Sensors, Motion Processing units, and pressure/contact sensors have been utilized for SLR in real-world scenarios, such




(a) (b)

**Fig. 1** Signs for fingerspelling of English alphabets (**a**) in the ASL [4], (**b**) in the ISL [5]

as harsh lighting and intricate backgrounds. These economical, lightweight sensors and the possibility to process the data on a low-power device such as an Arduino microcontroller make them viable solutions. However, as the number of gestures in a system increase, the chances of the being similar to another gesture also increase, which may affect the system's recognition accuracy.

Gestures are categorized based on whether they consist of movement of arms by the signer, called static and dynamic signs. Our paper deals with the recognition of static gesture signs. A significant problem with SL gestures is that the shape of one sign may be similar to that of another sign in terms of finger bend and wrist position; hence, the two signs may seem identical when done. This ambiguity might lead to incorrect categorization and low accuracy. Another problem is signer reliance. Because an individual cannot maintain all of their hands/fingers, the same differences in the capture of values can be seen by repeating the same sign many times.

Hence, the major contributions of this paper are as follows:

- Transfer learning-based vision-based system Effi-CNN proposed for recognition of SL gestures.
- A framework is proposed for SL gesture recognition to assist the deaf and hard-of-hearing people
- Experimental results indicate improved performance on five different SL datasets as compared to the existing SLR systems.

The remainder of this work is divided into the following sections. Section 2 introduces some related work in the field; Sect. 3 describes the system's components. Section 4 provides the exact implementation details of our model; Sect. 5 covers the datasets used and a comparative analysis of the proposed model's performance to those reported in the literature. Section 6 covers some discussion about the practical aspects of the SLR process—finally, Sect. 7 details the conclusions and future work.

## 2 Related work

Some of the existing work in literature in the domain of SLR using vision-based techniques are shown in Table 1.

The research gaps and learnings from the above Table 1 can be summarised into the following points:

1. Transfer learning is an effective method for SL classification problems since it provides superior performance and decreased training time since these models are already trained for superior performance on much larger classification problems of a similar category.
2. If the weights of pre-trained models are frozen, the models tend to be less accurate on the specific task since the models are more generalized.
3. Using RoI image segmentation results in improved classification accuracy due to removing unwanted features from the data and reducing training time.
4. Data augmentation has proven to be a highly effective method for making the model more robust to noise and improving overall performance.
5. Deep learning models are more accurate than machine learning models for image classification tasks.

**Table 1** Existing vision based techniques for Sign Language Recognition

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [29] | In this paper, authors have performed data augmentation by flipping, rotating (0°−30°), and brightness changes on the images. The image features were extracted using pre-trained CNN models ResNet50, MobileNet-V2 architectures, and a combination of the two | ResNet50, MobileNet-V2 | Trained for less epochs, leading to reduced performance. Accuracy increase was observed by using a combination of both models | ArSL 2018 [38, 39] |
| [30] | To overcome issues with classes containing different sample sizes, they used Synthetic Minority Oversampling Technique, after which their model's accuracy increased. After normalization and standard scaling, custom CNN model with 7 convolution layers was used for classification | CNN | Synthetic oversampling works better than minority oversampling, and minority under-sampling, showcasing the advantages of using data augmentation | ArSL 2018 |
| [31] | After image pre-processing and data normalization, the data was passed to a CNN model with 4 convolution layers. Compared to system trained on reduced size database, using the complete database, and training for more epochs, the testing accuracy increased to 97.6%, however, this significantly increased the training time by almost 9 times | CNN | Increasing the number of layers in the CNN model increased significantly; however, the number of images used for the training process didn't greatly impact the model's performance | ArSL 2018 |
| [32] | After resizing images to 32*32, data augmentation using horizontal flipping of the images, data normalization, and removing noisy images, 20,227 and 12,480 images were used for training and testing, respectively, of two custom CNN models with different layers | CNN | In their implementation, they only used 20,227 for the training set, had they used about 40,000 images from the overall set, their overall accuracy might have been better than 96.6% | ArSL 2018 |
| [33] | To overcome the imbalanced number of images in classes, they used only 1000 randomly selected images from each class before performing data augmentation and normalization. The data was then classified using an EfficientNetB4 based CNN classifier trained for 30 epochs | EfficientNetB4 | Their system took 10 h to train the model, which is considerably high; still, they attained an accuracy of 95%, which is low compared to other models in the study | ArSL 2018 |

**Table 1** (continued)

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [34] | They used a pre-trained VGG16 model for feature extraction from the images before passing it to fully connected dense layers of the CNN. GridSearchCV was used to find optimal values for hyperparameters tuning | VGG16 | This dataset achieved a considerably low accuracy of 94.33% compared to other models | ArSL 2018 |
| [35] | They used Sobel operator to find the 'absolute gradient magnitude' of the images for segmentation of the input images before passing to a 3 different transfer learning-based models, out of which VGGNet achieves the highest classification accuracy | VGGNet, GoogleNet, ALexNet, | Freezing weights of the pre-trained models led to the model not being trained for the actual dataset; otherwise, the accuracy of their system might have been better than 97% | ArSL 2018 |
| [36] | Initially, they resized the images to 28*28 and then applied the Gabor filter with 4 different sizes and 4 orientations to obtain spectral representations, which were then passed to a CNN classifier for recognition | CNN | The usage of Gabor filters increased the dataset by a factor of 16, which led to a high recognition accuracy of 99.05 on this dataset | ArSL 2018 |
| [37] | After under-sampling to equalize the number of images per class, they performed data augmentation before passing it to pre-trained VGG16, and ResNet152 models and trained for 100 epochs with the original weights frozen | VGG16, ResNet152 | Even though it attained a validation accuracy of 99.6%, it increased by only 0.1 percent from epoch 60 to 100 | ArSL 2018 |
| [40] | After image pre-processing, data augmentation, and standard scaling, the data was passed to a VGG-Net16 model for training model. The authors only trained their model for very few epochs because the pre-trained model's converged very rapidly | VGG16 | Their model underperforms for '0', 'N', 'W' gestures; if the model were trained for more epochs, it might have improved the classification accuracy | Massey [50, 51] |
| [41] | In this paper, a pre-trained Google Net architecture trained on the ILSVRC2012 dataset has been used for sign gesture classification | GoogleNet | The model's scalability remains a concern as their model provides validation accuracy of 98% with 5 letters and 75% with 10 letters | Massey |

**Table 1** (continued)

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [42] | The input data was passed to 3 models: A custom CNN model, pre-trained VggNet16 model, and Inception-V3 model. It was found that the pre-trained models outperform the custom CNN model by ~ 4%. Further, by using data augmentation, accuracy increased by ~ 3.5% | VGG16, Inception-V3 | Since the images were passed to 2 different models, and the fused feature set was passed to PCA for feature extraction, the selected features from the fused set resulted in high recognition accuracy of 99.63% | Massey |
| [43] | Single Shot Multibox Detector was used to detect and crop hand region before background subtraction on cropped image. 10 different sets of random background images were added to make the classifier more robust. The pixel values were then rescaled to be in 0–1 range, and passed to a Wide Residual Network (WRN) classifier | WRN-CNN | The performance increase of pre-trained WRN over a baseline CNN model further signifies the merits of pre-trained models trained over large datasets | Massey |
| [44] | This paper uses a multi-channel CNN-based feature fusion technique as a classifier. The input image is passed to one pipeline of the CNN as it is, and a Gabor filter is applied to the image before passing it to the second CNN pipeline for feature extraction | CNN | Ensemble model for feature extraction removes the high frequency spatial dependencies usually found in models trained with RGB images only and improves overall performance | Massey |
| [45] | They used Media-pipe hands API to extract coordinates of 21 joint points on the hand of the signer. Then distances between the sets of these joint points were calculated for a total of 190 distance vectors; similarly, 210 angle vectors were calculated using the relative distance between the joint points. Also, angle-based vectors were calculated to fetch the hand's orientation. All these feature vectors were then passed to a SVM and a light Gradient Boosting Machine (GBM) for training and classification. Testing accuracies of 99.39% and 97.80% were obtained for SVM, and GBM | SVM, GBM | Their system achieves a per sample recognition time of only 14 ms, which is highly adequate for a real-time SLR system given that their system also achieves high recognition accuracy of 99.39% on this dataset and, respectively, high accuracies on 2 more ASL datasets | Massey |

**Table 1** (continued)

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [46] | They used a RCNN-based model to detect hand regions in the image, from which 5 different regions were cropped out. After this, Gaussian and Salt-and-pepper noise were added to the images in 4 different varying proportions. The original image, cropped images, and noise-added cropped images were passed to 3 different Restricted Boltzmann Machines, whose outputs were fused for final output generation | RCNN | Multi-channel data fusion and multiple transformations of each image results in high accuracy of 99.31%. The drawback with their system is that it was run for 10,000 epochs and the high design complexity | Massey |
| [47] | YCbCr based hand region segmentation and Convex Hull Algorithm based hand shape detection for image cropping was used before passing to a CNN classifier | CNN | Hand location-based segmentation reduces useless features and training time, as well as increases accuracy | Massey |
| [28] | VGG16 for classification, GridSearchCV for optimal hyperparameters tuning | VGG16 | Model's lower performance w.r.t. to others models can be due to the absence of specified feature extraction steps | Massey |
| [48] | After cropping and resizing images based on skin segmentation, they used Local-Binary-Pattern (LBP), Histogram-of-Oriented-Gradients (HOG) as feature extractors, and CNN, SVM as a classifier. They ran multiple combinations of these classifiers and feature extractors | CNN, SVM | The HOG-LBP-SVM model provides better accuracy than the CNN-SVM and other models, however, increased implementation complexity remains a concern | Massey |
| [49] | The authors propose a SLR system based on CNN models built using transfer learning from image multiple classifier systems. The input images will be resized, and background subtraction and convex hull algorithm for image processing will be used | CaffeNet, GoogleNet, AlexNet, VGGNet | Only a model is proposed for future implementation and does not include any implemented model | Massey |

**Table 1** (continued)

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [52] | In this paper, sign images were acquired from the video feed, and after pre-processing to extract the hand region and convert the RGB image to HSV, the CNN model was used for training and classification | CNN | Their model converges in only 5 epochs with a high learning rate of 0.1; it can be assumed that model's accuracy of 98.55% can increase further with an improved learning rate | ASL [52, 54] |
| [53] | In this paper, the authors have used the same technique as in [52]; however, they achieve better test accuracy on the dataset by using a different configuration of the CNN layers. They further evaluated their model's performance on 2 more datasets, as shown | CNN | The increasing number of epochs to 100 allowed them to get increased accuracy of 99.41% on the dataset [52], 99.48% on the dataset [55], and 99.38% on their own synthetic dataset [56] | ASL [52]<br>ASL [55]<br>ASL [56] |
| [57] | Human skin colour based hand portion segmentation using HSV values was done before passing the images to a SVM classifier for training and classification. The system works on mobile phones, making it available to the masses for real-time SLR | SVM | However, the system's accuracy is only 80–90 percent for different alphabets, which makes it inadequate compared to other systems in our study | ASL [57] |
| [58] | A Region Proposal Network was used for ROIs in the image then pooling was used to bring all those segments to the same size before passing it to the CNN classifier | CNN | Using an attention-based model to detect the hand region in the image reduces the processing load for the classification part and increases accuracy by 5% | ASL [64] |
| [59] | After converting the RGB images to grayscale and using the Canny edge detector for edge detection [60], the image was passed to multiple feature extractors, including HOG, LBP, PCA, and ORB [61] feature detector, which produces a 32-dimension feature vector. After converting these features to a bag-of-words [62] model, multiple machine learning-based classifiers were used to train and classify the gestures | SVM, logistic regression, Naïve Bayes, KNN, Random forest, MLP, | A combination of PCA with a multi-layer-perceptron (MLP) classifier achieved highest classification accuracy | ASL [64] |

**Table 1** (continued)

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [60] | They used multiple CNN based classifiers using transfer learning methods. It was found that the pre-trained models offer much better classification accuracy than custom CNN models, with the ResNet50 model achieving highest accuracy | VGG16, InceptionNet, and ResNet50 | This paper found that the performance of InceptionNet was lower than baseline CNN model, and Resnet50 outperformed VGG16 | ASL [64] |
| [61] | After resizing, the image was broken down into smaller patches. These patches were passed to the Vision Transformer (ViT) encoder consisting of the attention layer and MLP. The output of this layer was then passed to another MLP for final classification | MLP | Due to the low accuracy of 80.6%, this model cannot be considered an adequate solution for the problem under consideration | ASL [64] |
| [62] | This paper used image resizing to 50*50 pixels, data augmentation using Gaussian noise adding and image rotation. The images were then passed to a CNN model with 4 convolution layers for training and classification | CNN | Their model provides excellent testing accuracy of 99.89% but takes 0.236 s per image for recognition, which makes it unsuitable for a real-time system | ASL [64] |
| [63] | This paper tested the transfer learning model for gesture classification using VGG16 model with and without Bimodal Distribution Removal (BDR). It was found that the use of BDR leads to significant improvement in recognition accuracy | VGG16 | Only 9% of images from the dataset were used for training, and they got 68% testing accuracy. Hence the results obtained cannot be extrapolated to the entire dataset | ASL [64] |
| [42] | A combination of VggNet, and Inception-V3 were used for this dataset too | VggNet, Inception-V3 | 99.02% accuracy on dataset [64] | ASL [64] |
| [65] | Image data was passed to two pre-defined models with and without augmentation. It was observed that LeNet obtained lowest accuracy; accuracy increased by 6.74% in the CapsNet model fed with the original dataset, while accuracy increased by another 6.62% using augmented data | LeNet, CapsNet | The results in this paper further strengthen the case that data augmentation used during training can increase the system's overall performance | MNIST [71] |

**Table 1** (continued)

| Ref | Methodology | Classifier | Conclusion | Dataset |
|---|---|---|---|---|
| [36] | Gabor filters with CNN | CNN | 99.9 accuracy on MNIST dataset signifies the advantage of using Gabor filter for vector creation for image classification | MNIST |
| [66] | A custom CNN model was proposed with 3 convolution layers and ran for 10 epochs over the training set | CNN | The system was evaluated using only 10 images, which cannot be extrapolated for complete dataset | MNIST |
| [67] | In this paper, numeric data from xls files of the dataset were converted to PNG files and then passed to two models: MobileNet and InceptionV3. These models were trained, and then the trained model was used for gesture classification on a smartphone | MobileNet, InceptionV3 | The average recognition time for a gesture was about 2.42 s, which implies that the model was not sufficiently optimized to run on low-performance hardware | MNIST |
| [68] | In this paper, Particle Swarm Optimisation (PSO) was used to optimize the parameters affecting the classification accuracy of a CNN-based image classifier. The use of PSO results in an accuracy of 99.53% | CNN | The time required for the system to run for all the combinations of parameters is quite long and not scalable for large datasets | MNIST |
| [69] | This model uses a CNN classifier incorporating 7 convolution layers and 2 dense layers for image classification | CNN | The model's accuracy can be considered to be low | MNIST |
| [70] | In this paper, discrete wavelet transform was used to extract features from the images, to be classified by the CNN module | CNN | Using wavelet transform to extract low-level features from images results in high accuracy | MNIST |

6. The classification accuracy improves by increasing the number of images per class to train the models.
7. After a certain high level of accuracy has been achieved, increasing the complexity of the model to suit the task better provides diminishing improvements.
8. In the cases where the number of epochs for which the model is trained is deficient (less than 20), the model usually did not learn enough features from the data to classify the test images accurately.

## 3 Preliminary

In order to recognize Sign language gestures [72], transfer learning [73] using modified pre-trained EfficientNetB2 [74] based CNN [75, 76] models have been used. The architecture of these elements has been discussed in the following sections. Also, detailed information used in our proposed implementation, like OpenCV [77], Keras [78], TensorFlow [79] etc., have also been discussed below.

a) Sign Language: It is the way of communication by using hand gestures, body language, and facial expressions. It is the main form of contact for the Deaf and Hard-of-Hearing community. SL gestures are categorized based on the movement type as static gestures that remain invariant to motion for the period of the gesture and dynamic gestures that are centred on movement. Signs can be performed using one hand or both hands, static or dynamic, and other categorizations of signs are possible, as shown in Fig. 2. Figure 2, the type 0 sign indicates a two-handed dynamic sign in which both hands are active. Type 1 is a two-handed dynamic sign with only the primary hand performing the motion.

b) Convolution neural networks are one of the most widely used deep learning approaches for visual imagery analysis. In comparison to other image classification techniques, CNN requires less pre-processing. CNN is a feed-forward artificial neural network comprising many neural network layers, each having multiple neurons. The network learns feature extraction filters that are hand-modelled in other paradigms. Convolution, pooling, flattening, and fully linked layers are the four types of processes in a CNN. The convolutional layer frequently captures low-level characteristics like colour, edges, and gradient direction. The pooling layer reduces the spatial dimension of the convolved features. This process minimizes the needed computing time for working with the data through dimensionality alleviation. It also has the benefit of keeping dominating characteristics throughout model training, which are invariant in position and rotation. Higher-level characteristics can be categorised after the input picture has been processed. As a result, the picture is flattened into a one-dimensional vector. The flattened output is sent into a fully connected layer in CNN, which classifies the features retrieved by the convolutional/pooling layer into labels. The model may offer the probability of predicting items in the picture after training using SoftMax [81] classification.

c) Transfer Learning. Transfer learning [73] is a procedure that influences pre-trained models that are trained on an enormous and miscellaneous dataset to solve novel, related tasks with partial data. Transfer learning has assisted attain achievement in deep learning, with fine-tuning a pre-trained model providing an unlimited way for knowledge transfer. The process of transfer learning involves leveraging the knowledge gained from a source domain and applying it to a target domain. This is done by

reusing the learned features, such as weights and biases, from the pre-trained model and fine-tuning them to fit the new task. This approach reduces the need for large amounts of labeled data and decreases training time, while still attaining high performance. By utilizing pre-trained models, transfer learning can effectively handle domain adaptation, allowing the model to perform well on the target task.

d) EfficientNet. CNNs are frequently built at a fixed resource cost and then scaled up when more resources become available to improve accuracy. However, traditional model scaling strategies are extremely unpredictable. Some models are scaled in depth, while others are scaled in width. Some models simply use higher-resolution photos to achieve better outcomes. This method of arbitrarily scaling models necessitates manual adjustment and many man-hours, with little or no performance increase. EfficientNet scales up models using a simple yet effective compound coefficient approach. Compound scaling consistently increases each dimension with a predetermined set of scaling coefficients rather than randomly scaling up width, depth, or resolution. The authors created seven models [B0 to B7] of varied dimensions, surpassing most CNNs' performance and efficiency [82, 83].

## 4 Proposed system

Our proposed system consists of a pre-trained model, EfficientNetB2. Several design considerations were taken to get to the final architecture. Decisions on the pre-trained model, optimization techniques, and hyper-parameters to test and assess are more specific. After the base model, we added two dense layers at the end for classification. Weights of the base model pre-trained on the Imagenet dataset have been chosen in the model; also, the weights of the base model were not frozen and the model was fine-tuned on the dataset for domain adaptation. The optimizer used is Adamax with loss as categorical cross-entropy, a learning rate of 0.001 with a learning rate reduction factor of 0.5, and a momentum of 0.99. Dwell is set as true so that if the monitored metric does not improve on the current epoch, then set model_weights back to weights of the previous epoch. The include_top parameter was set as false to fetch outputs activation from the last pooling or convolution layer of the pre-trained model and feed them to a dense layer with 256 neurons, followed by a dropout layer with a dropout rate of 0.45, followed by another dense layer with softmax activation and several neurons equal to the number of classes in the input dataset. In total, the final model contains 344 layers and 8,144,414 Trainable parameters.

In this study, we also developed an application for real-time SLR. Images from webcam footage were captured using the OpenCV library. During runtime, three windows appear when the application is in real-time recognition mode. The first window is the picture capture window, which has a 192*192 pixel predetermined rectangle; the user must execute the signs inside the given area, as shown in Fig. 3 (a).

After capturing the picture from the predefined region, the cropped image is converted to HSV and displayed in the preview window (shown overlaid on the preview window) using user-defined parameters during runtime by using a third window. Then, the application supplies it to the pre-trained CNN model for classification. This study's model was trained using augmented data from multiple ASL datasets. Finally, the anticipated character appears in the image capture window's specified region. A third control window as shown in Fig. 3 (b) with track bars, is used for controlling the HSV threshold

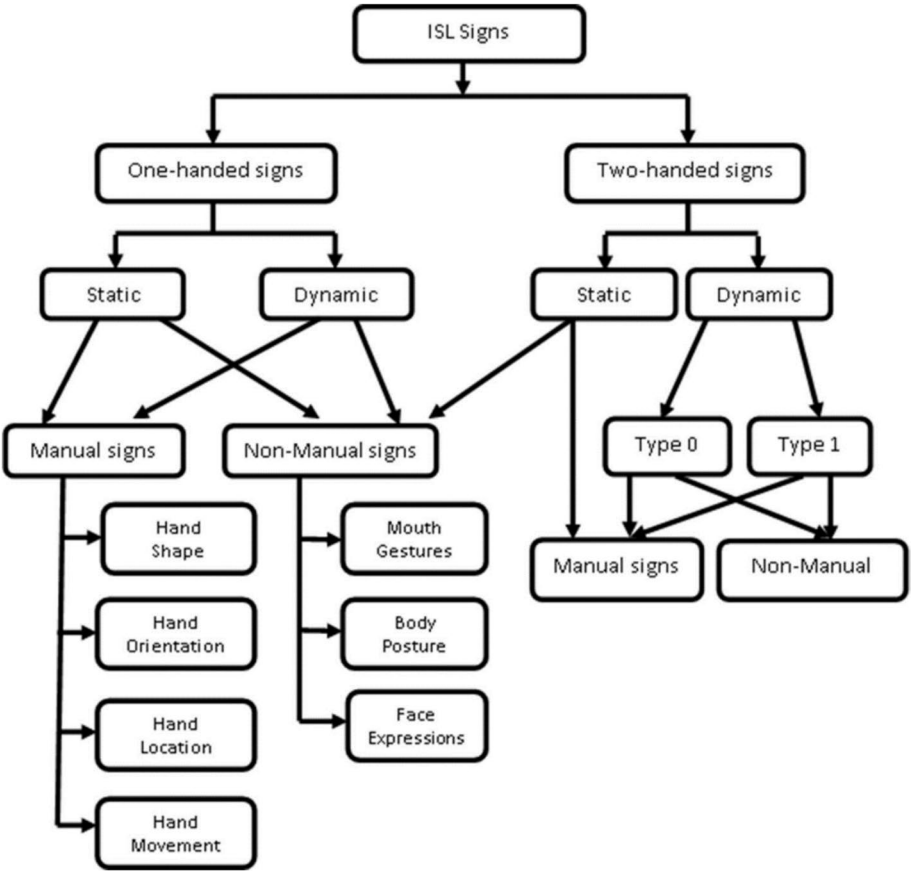



**Fig. 2** Systematic representation of the Hierarchy of sign language gestures [80]

for skin colour based feature extraction during runtime; this is needed since, in different lighting conditions, skin colour appears different and appears to be of a different hue. Figure 4 has the following overall steps:

i. Collect and pre-process the data.
ii. Split the dataset into training and validation sets.
iii. Transfer Learning model: Initialize the EfficientNetB2 model with pre-trained weights on the image dataset. Then, freeze all the layers except the last few layers to prevent overfitting on the small dataset and replace the last layer with a new fully connected layer with the appropriate number of output classes.
iv. Fine-tuning: Train the model on the training set using an appropriate optimizer and loss function. Monitor the model's performance on the validation set and adjust hyper-parameters such as learning rate and batch size as needed.
v. Train and evaluate the model.

The data flow diagram depicting the training process of the system is shown in Fig. 5, while the data flow of the real-time SLR system is shown in Fig. 4.

## 5 Experiment analysis

This section describes the SLR datasets in our implementation and implementation details.

### 5.1 i. Datasets

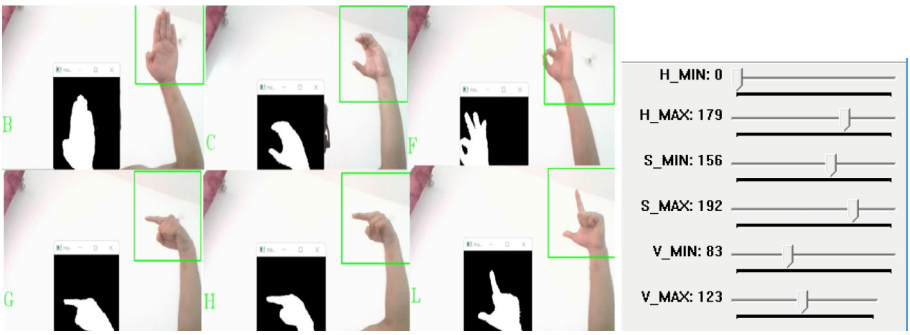
Our experiments are carried out on eight different Sign language datasets, including six ASL datasets and one Arabic and Pakistani SL dataset. Details about these datasets are shown in Table 2. If the train and test sets were not already in segregated form for all the datasets, then the data was split in an 80–20 ratio for training-test sets, respectively. After this, for all the datasets, the training set was split into a 95–5 ratio for training and validation data, respectively.

### 5.2 ii. Dataset limitations

1) In most cases, the whole datasets were recorded with the same background conditions at the same place.
2) There is insufficient orientation, posture, noise, lightning, and zooming variations in the dataset's classes.
3) The number of signers involved in dataset creation was usually very low.
4) In some datasets, the number of images differs from one class to another, e.g. in [38, 71].

### 5.3 iii. Classification prediction results and performance comparison with other models

Eight separate SL datasets were used in the evaluation process of our proposed model. The details of these datasets are presented in the methodology section. If the train data




(a) (b)

**Fig. 3** **a** First two windows of the application, (**b**) HSV parameters control trackbars window

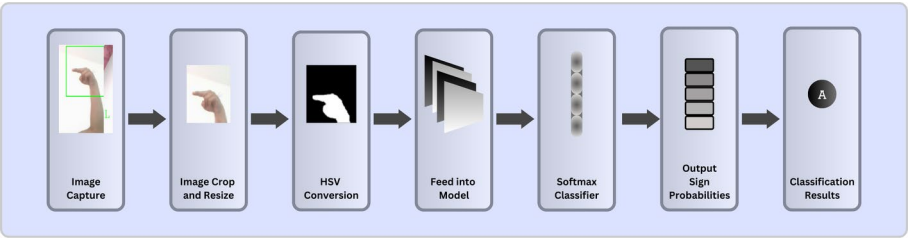



**Fig. 4** Data flow model of the proposed system for real-time testing of the system

set is extensive, we have chosen to trim the training dataset to 1000 samples per class and the validation dataset to 50 samples per class to reduce training time. All the images were resized to 50*50 pixels before feeding to the model. According to the results of the experiments, our proposed Effi-CNN model outperformed almost other existing models in test accuracy. The performance of our model on the datasets, as mentioned earlier, is shown below in Figs. 6, Fig. 7, Fig. 8, and Fig. 9. A comparison of Effi-CNN with the best approach in the literature and training time taken is presented in Table 3.

Compared to other systems, our system takes much less time for the training process because the model's weights did not have to be trained from scratch, resulting in a faster convergence rate.

The training was halted during our training process when validation accuracy increased beyond 99.5% to avoid overfitting.

## 6 Discussion

I. Processing Time: Our model was run on a Google Colab online execution platform for training and testing the datasets. The time required for training our system for all the datasets is specified in Table 3. In practical applications, the processing time taken for gesture recognition is essential for a better user experience. In our system, we have developed a real-time application to capture gesture input using a webcam and perform some pre-processing on the RAW image before sending it to pre-trained CNN model for classification. The processing delays were found to be within a tolerable range of 0.01 s on an 'AMD A8-7410 APU Quad Core 2.2 GHz system, with 4 GB DDR-3 RAM running at 1200 MHz', which is adequate for a real-time system. It is worth noting that only one CPU core is activated during the recognition process, with about 25 percent load on the core.

II. Skin Tone Impact: In our application, in the pre-processing stage, the RGB image acquired from the camera is converted using HSV filter to extract only human skin coloured parts from the image. However, people have a wide variety of skin tones, and various skin tones absorb light differently, affecting the accuracy of gesture detection. As a result, different HSV values were required for successful segmentation in various illumination scenarios.

III. Impact of Intense Body Movements: During the real-time gesture recognition process, the recognition rate of the system is limited by the frame rate of the camera in consideration, which is usually 24–30 fps on a laptop webcam. If abrupt motions are made, captured images might be blurred due to motion artefacts, which leads to distorted image, which is rendered useless for the recognition process.

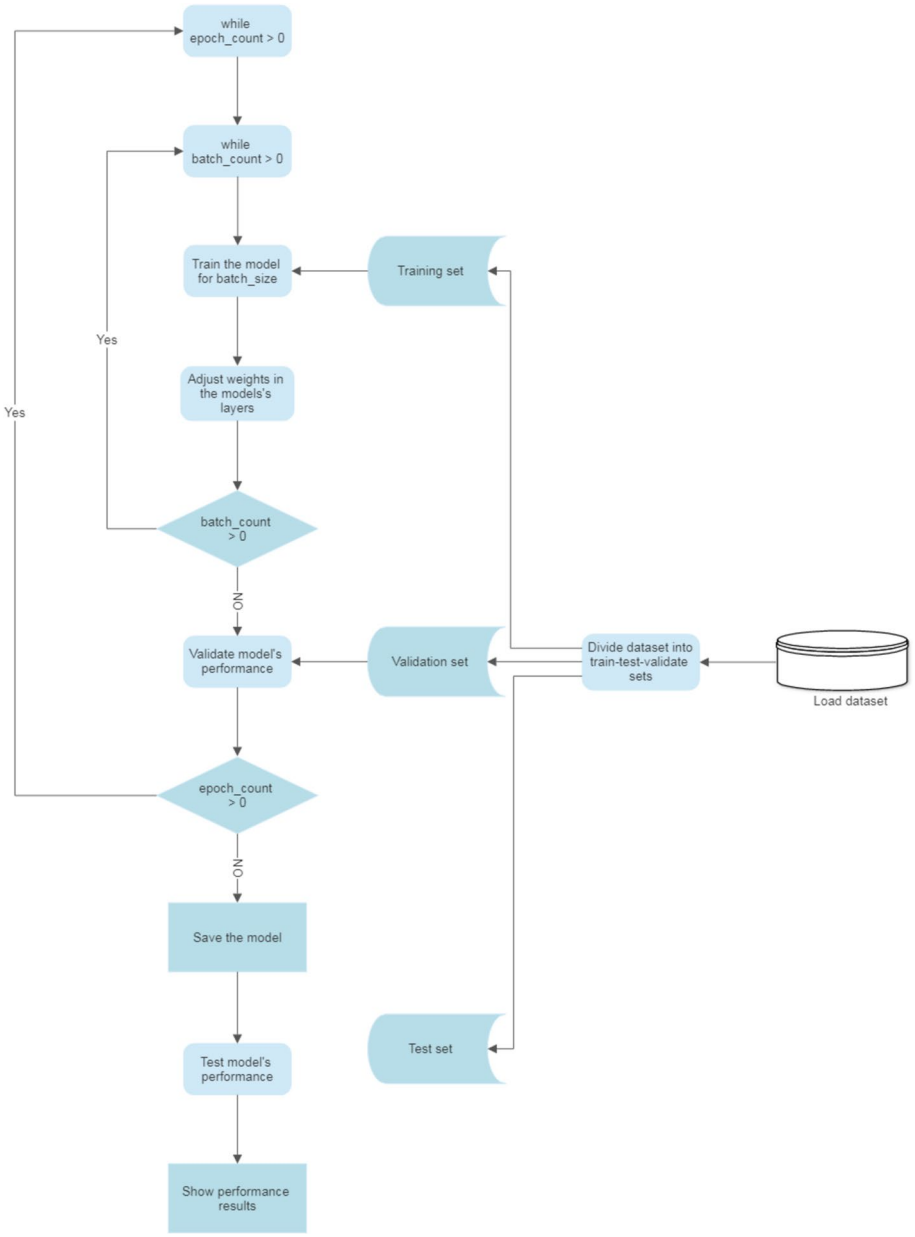



**Fig. 5** Data flow model of the proposed system for training and testing accuracy of the system

IV. Background and lighting conditions: Both background colour and lighting conditions play a crucial role in image classification tasks. If the background is uniformly coloured and does not match the signer's skin colour, extracting the hand segment from the image is more accessible. If any skin-color objects are present in the ROI, the system may fail to segment the hand area from the image.

**Table 2** Details of datasets included in our study

| Dataset | Sign Language | Type of dataset | Vocabulary size | Number of Signers | Total number of images |
|---|---|---|---|---|---|
| ArSL [38, 39] | Arabic SL | Grayscale Images | 32 | 40 | 54,049 |
| Massey [50, 51] | American SL | RGB Images | 36 | 5 | 2,425 |
| [52, 54] | American SL | HSV-BW Images | 26 | 1 | 52,000 |
| [55] | American SL | HSV-BW Images | 44 | 1 | 105,600 |
| [56] | American SL | HSV-BW Images | 26 | 1 | 104,000 |
| PSL [57] | Pakistani SL | RGB Images | 37 | NA | 1,509 |
| [64] | American SL | RGB Images | 29 | NA | 87,000 |
| MNIST [71] | American SL | CSV file | 24 | NA | 27,455 |

*NA* data not provided

In dim light situations, the captured images contain more noise than those captured in well-lit conditions. This leads to poor edge detection and error prone segmentation, which leads to poorer recognition accuracy.

## 7 Conclusions and future work

A hand gesture recognition approach for vision-based sign language recognition is provided in this study. A deep learning-based Effi-CNN model based on transfer learning using EfficientNetB2 as the base model is presented for vision-based static gesture

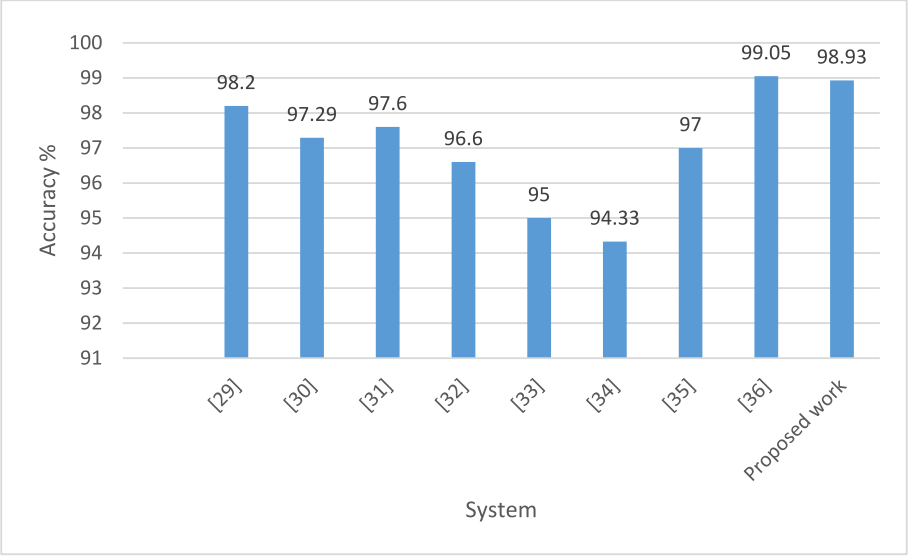



**Fig. 6** Performance of different models on ArSL dataset

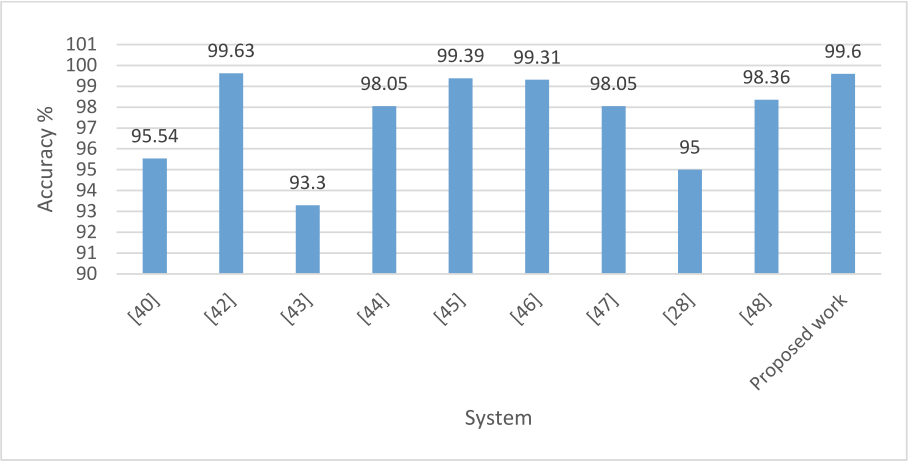



**Fig. 7** Performance of different models on Massey University dataset

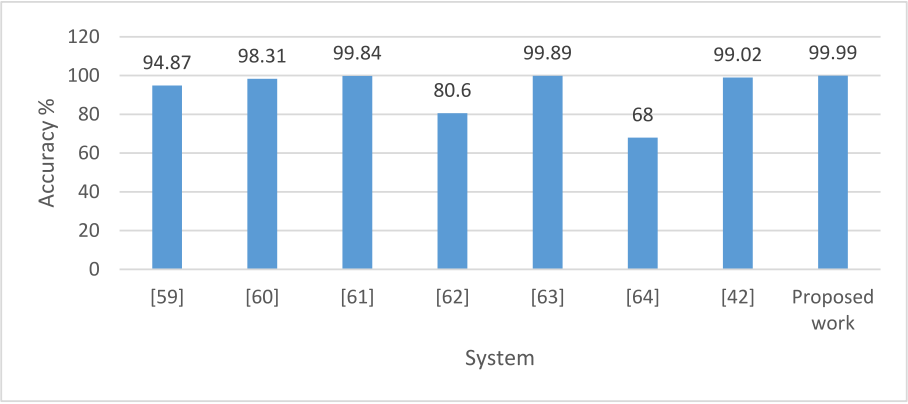



**Fig. 8** Performance of different models on dataset [64]

identification. External gear such as Kinect sensors is not required for the proposed vision-based paradigm, making it more feasible. The capacity of this research to recognize the indicators of distinct SL with good recognition results over state-of-the-art techniques is a noteworthy addition. The performance of this work is assessed using 8 publically available SL datasets. According to rigorous experimental assessment, the Effi-CNN model either outperforms most existing systems or produces competitive results compared to other state-of-the-art approaches.

The model attempts to classify the input image based on relevant trained patterns learned during training by calculating the probability of a class being most likely. To correctly translate hand-sign-spelled words and sentences into written language, we must continue to create future work that includes all punctuation marks and all types of joint letter representation. This study, on the other hand, will serve as a point for researchers working in the field of SLR. The system may be used as a translator for sign language and non-sign language communication, as well as for human–computer/machine interface and robot

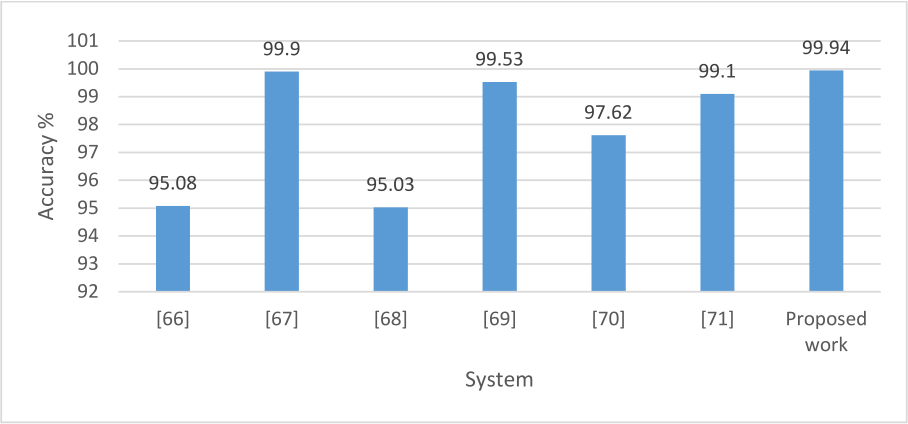



**Fig. 9** Performance of different models on the MNIST dataset

**Table 3** Training details of Effi-CNN on the datasets and comparison with the best approach present in literature

| Dataset | Accuracy of our system | Best accuracy in literature | Training time |
|---|---|---|---|
| MNIST [71] | 99.88—99.94 | 99.90 | 12 min, 24 s |
| PSL [57] | 99.34 – 99.60 | 80–90 | 26 min, 48 s |
| ArSL2018 [38] | 98.93 | 99.05 | 44 min, 33 s |
| Massey [50] | 99.45 – 99.60 | 99.63 | 24 min, 8 s |
| [64] | 99.92 | 99.89 | 59 min, 11 s |
| [52, 54] | 99.49 – 99.61 | 99.41 | 33 min, 12 s |
| [55] | 99.62 | 99.48 | 59 min, 54 s |
| [56] | 99.29 – 99.31 | 99.38 | 36 min, 31 s |

control. Currently, our work focuses on static gesture recognition, and we also intend to recognize dynamic sign language gestures through video frames, which is a challenging task. Also, our present method is based on a definite zone of interest that has been designated. There would be no need for a predefined ROI if object detection was used to find the hand region, which is a must for practical applications. If the text output of the system can be translated to voice using a text-to-speech application such as Google's Cloud Text-To-Speech for a better experience.

**Data availability** Not applicable to this article as no datasets were generated or analyzed during the current study'.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References


1. National Institute on Aging. n.d. Hearing Loss: A Common Problem for Older Adults.[online] Available at: <https://www.nia.nih.gov/health/hearing-loss-common-problem-older-adults> [Accessed 24 May 2022].
2. Disabled World. (2022, April 7). Deaf Communication: Sign Language and Assistive Hearing Devices. Disabled World. Retrieved May 24, 2022 from www.disabled-world.com/disability/types/hearing/communication/
3. Ai-Media creating accessibility, one word at a time. 2021. Sign Language Alphabets From Around The World - ASL - Ai-Media. [online] Available at: <https://www.ai-media.tv/ai-media-blog/sign-language-alphabets-from-around-the-world/> [Accessed 24 May 2022].
4. En.wikipedia.org. Fingerspelling - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Fingerspelling> [Accessed 28 April 2022].
5. Islrtc.nic.in. Poster of the Manual Alphabet in ISL | Indian Sign Language Research and Training Center (ISLRTC), Government of India. [online] Available at: <http://www.islrtc.nic.in/poster-manual-alphabet-isl> [Accessed 26 April 2022].
6. Lifeprint.com. n.d. American Sign Language (ASL). [online] Available at: <https://www.lifeprint.com/asl101/pages-layout/grammar.htm> [Accessed 24 May 2022].
7. Ai-Media creating accessibility, one word at a time. 2022. Sign Language Alphabets From Around The World - ASL - Ai-Media. [online] Available at: <https://www.ai-media.tv/ai-media-blog/sign-language-alphabets-from-around-the-world/> [Accessed 27 May 2022].
8. Lackner A (2017) Comprehending the Complexities of Sign Language. [online] The Wire. Available at: <https://thewire.in/culture/amazing-complexity-sign-language> [Accessed 24 May 2022].
9. Li B, Yang J, Yang Y, Li C, Zhang Y (2021) Sign Language/Gesture Recognition Based on Cumulative Distribution Density Features Using UWB Radar. IEEE Trans Instrum Meas 70(1–13):2511113. https://doi.org/10.1109/TIM.2021.3092072
10. Yang Y, Li J, Li B et al (2022) MDHandNet: a lightweight deep neural network for hand gesture/sign language recognition based on micro-doppler images. World Wide Web. https://doi.org/10.1007/s11280-021-00985-1
11. Wang P, Zhou Y, Li Z, Huang S, Zhang D (2021) Neural Decoding of Chinese Sign Language With Machine Learning for Brain-Computer Interfaces. IEEE Trans Neural Syst Rehabil Eng 29:2721–2732. https://doi.org/10.1109/TNSRE.2021.3137340
12. Malaia EA, Borneman SC, Krebs J, Wilbur RB (2021) Low-Frequency Entrainment to Visual Motion Underlies Sign Language Comprehension. IEEE Trans Neural Syst Rehabil Eng 29:2456–2463. https://doi.org/10.1109/TNSRE.2021.3127724
13. AlQattan D, Sepulveda F (2017) Towards sign language recognition using EEG-based motor imagery brain computer interface. 2017 5th International Winter Conference on Brain-Computer Interface (BCI), 5–8, https://doi.org/10.1109/IWW-BCI.2017.7858143.
14 Kubicek E, Quandt LC (2019) Sensorimotor system engagement during ASL sign perception: An EEG study in deaf signers and hearing non-signers. Cortex 119:457–469. https://doi.org/10.1016/j.cortex.2019.07.016
15 Zhang L, Zhang Y, Zheng X (2020) WiSign: Ubiquitous American Sign Language Recognition Using Commercial Wi-Fi Devices. ACM Trans Intell Syst Technol 11(3):24. https://doi.org/10.1145/3377553
16. Shang J, Wu J (2017) A Robust Sign Language Recognition System with Multiple Wi-Fi Devices. In Proceedings of the Workshop on Mobility in the Evolving Internet Architecture (MobiArch '17). Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/3097620.3097624
17. Shang J, Wu J (2017) A Robust Sign Language Recognition System with Sparsely Labeled Instances Using Wi-Fi Signals. 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 99–107, https://doi.org/10.1109/MASS.2017.41.
18 Ahmed HFT, Ahmad H, Narasingamurthi K, Harkat H, Phang SK (2020) DF-WiSLR: Device-Free Wi-Fi-based Sign Language Recognition. Pervasive Mob Comput 69:101289. https://doi.org/10.1016/j.pmcj.2020.101289
19 Gurbuz SZ et al (2021) American Sign Language Recognition Using RF Sensing. IEEE Sensors J 21(3):3763–3775. https://doi.org/10.1109/JSEN.2020.3022376
20. Amrutha K, Prabu P (2021) ML Based Sign Language Recognition System. Int Conf Innov Trends Inform Technol (ICITIIT) 2021:1–6. https://doi.org/10.1109/ICITIIT51526.2021.9399594
21. Zhang LG, Chen Y, Fang G, Chen X, Gao W (2004) A vision-based sign language recognition system using tied-mixture density HMM. In Proceedings of the 6th international conference on Multimodal

interfaces (ICMI '04). Association for Computing Machinery, New York, NY, USA, 198–204. https://doi.org/10.1145/1027933.1027967
22. Karami A, Zanj B (2011) Azadeh Kiani Sarkaleh, Persian sign language (PSL) recognition using wavelet transform and neural networks. Expert Syst Appl 38(3):2661–2667. https://doi.org/10.1016/j.eswa.2010.08.056
23. Wang CC et al (2021) Real-Time Block-Based Embedded CNN for Gesture Classification on an FPGA. IEEE Trans Circuits Syst I Regul Pap 68(10):4182–4193. https://doi.org/10.1109/TCSI.2021.3100109
24. Sincan OM, Keles HY (2020) AUTSL: A Large Scale Multi-Modal Turkish Sign Language Dataset and Baseline Methods. IEEE Access 8:181340–181355. https://doi.org/10.1109/ACCESS.2020.3028072
25. Plouffe G, Cretu A (2016) Static and Dynamic Hand Gesture Recognition in Depth Data Using Dynamic Time Warping. IEEE Trans Instrum Meas 65(2):305–316. https://doi.org/10.1109/TIM.2015.2498560
26. Sun C, Zhang T, Bao B, Xu C, Mei T (2013) Discriminative Exemplar Coding for Sign Language Recognition With Kinect. IEEE Transactions on Cybernetics 43(5):1418–1428. https://doi.org/10.1109/TCYB.2013.2265337
27. Avola D, Bernardi M, Cinque L, Foresti GL, Massaroni C (2019) Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures. IEEE Trans Multimedia 21(1):234–245. https://doi.org/10.1109/TMM.2018.2856094
28. Rho E, Chan K, Varoy EJ, Giacaman N (2020) An Experiential Learning Approach to Learning Manual Communication Through a Virtual Reality Environment. IEEE Trans Learn Technol 13(3):477–490. https://doi.org/10.1109/TLT.2020.2988523
29. Alnuaim Abeer, Zakariah Mohammed (2022) Human-Computer Interaction with Hand Gesture Recognition Using ResNet and MobileNet. Comput Intell Neurosci 8777355(16):2022. https://doi.org/10.1155/2022/8777355
30. Alani, AA, Cosma G (2021): ArSL-CNN: a convolutional neural network for Arabic sign language gesture recognition. Loughborough University. Journal contribution. https://hdl.handle.net/2134/16878787.v1 https://doi.org/10.11591/ijeecs.v22.i2.pp1096-1107
31 Latif G, Mohammad N, AlKhalaf R, AlKhalaf R, Alghazo J, Khan M (2020) An automatic Arabic sign language recognition system based on deep CNN: an assistive system for the deaf and hard of hearing. Int J Comput Digital Syst 9(4):715. https://doi.org/10.12785/ijcds/09041824
32. Hayani S, Benaddy M, El Meslouhi O, Kardouchi M (2019) Arab sign language recognition with convolutional neural networks. In: 2019 International Conference of Computer Science and Renewable Energies (ICCSRE). Agadir, Morocco, pp 1–4. https://doi.org/10.1109/ICCSRE.2019.8807586
33 Zakariah M, Alotaibi YA, Koundal D, Guo Y, Mamun Elahi M (2022) (2022) Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique. Comput Intell Neurosci 4567989:15. https://doi.org/10.1155/2022/4567989
34. Nurnoby MF, El-Alfy ESM, Luqman H (2020) Evaluation of CNN Models with Transfer Learning for Recognition of Sign Language Alphabets with Complex Background. 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), 1–6, https://doi.org/10.1109/3ICT51146.2020.9311989.
35. Duwairi RM, Halloush ZA (2022) Automatic recognition of Arabic alphabets sign language using deep learning. International J Electr Comput Eng (2088–8708) 12(3)
36. Luqman H, El-Alfy ESM, BinMakhashen GM (2021) Joint space representation and recognition of sign language fingerspelling using Gabor filter and convolutional neural network. Multimed Tools Appl 80:10213–10234. https://doi.org/10.1007/s11042-020-09994-0
37. Saleh Y, Issa GF (2020) Arabic Sign Language Recognition through Deep Neural Networks Fine-Tuning. Int J Online Biomed Eng (iJOE) 16(05):71–83. https://doi.org/10.3991/ijoe.v16i05.13087
38. Latif G, Alghazo J, Mohammad N, AlKhalaf R, AlKhalaf R (2018) Arabic Alphabets Sign Language Dataset (ArASL). [online] Mendeley Data. Available at: <https://data.mendeley.com/datasets/y7pckrw6z2/1> [Accessed 24 May 2022].
39 Latif G, Mohammad N, Alghazo J, AlKhalaf R, AlKhalaf R (2019) ArASL: Arabic Alphabets Sign Language Dataset. Data in Brief 23:103777. https://doi.org/10.1016/j.dib.2019.103777
40. Masood S, Thuwal H, Srivastava A (2018). American Sign Language Character Recognition Using Convolution Neural Network. https://doi.org/10.1007/978-981-10-5547-8_42.
41. Garcia B, Viesca SA (2016) Real-time American sign language recognition with convolutional neural networks. Convolutional Neural Netw Visual Recognition 2:225–232
42. Rajan RG (2021) Transfer-learning analysis for sign language classification models. Turkish J Comput Math Educ (TURCOMAT) 12(9):1423–1433

43. Kania K, Markowska-Kaczmar U (2018). American Sign Language Fingerspelling Recognition Using Wide Residual Networks. https://doi.org/10.1007/978-3-319-91253-0_10.
44 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. https://doi.org/10.1016/j.asoc.2018.09.010
45. Shin J, Matsuoka A, Hasan MAM, Srizon AY (2021) American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation. Sensors 21(17):5856. https://doi.org/10.3390/s21175856
46. Rastgoo R, Kiani K, Escalera S (2018) Multi-Modal Deep Hand Sign Language Recognition in Still Images Using Restricted Boltzmann Machine. Entropy 20:809. https://doi.org/10.3390/e20110809
47. Taskiran M, Killioglu M, Kahraman N (2018) A Real-Time System for Recognition of American Sign Language by using Deep Learning. 2018 41st International Conference on Telecommunications and Signal Processing (TSP), 1–5, https://doi.org/10.1109/TSP.2018.8441304.
48. Nguyen HBD, Do HN (2019) Deep Learning for American Sign Language Fingerspelling Recognition System. 2019 26th International Conference on Telecommunications (ICT), 314–318, https://doi.org/10.1109/ICT.2019.8798856.
49. Salian S, Dokare I, Serai D, Suresh A, Ganorkar P (2017). Proposed system for sign language recognition. 058–062. https://doi.org/10.1109/ICCPEIC.2017.8290339
50. Massey.ac.nz (2012) Massey University. [online] Available at: <https://www.massey.ac.nz/~albarcza/gesture_dataset2012.html> [Accessed 24 May 2022].
51. Barczak ALC, Reyes NH, Abastillas M, Piccio A, Susnjak T (2011) A new 2D static hand gesture colour image dataset for ASL gestures. Res Lett Inform Math Sci 15:12–20 (http://iims.massey.ac.nz/research/letters/)
52. Jeny JRV, Anjana A, Monica K, Sumanth T, Mamatha A (2021) Hand Gesture Recognition for Sign Language Using Convolutional Neural Network. 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), 1713–1721, https://doi.org/10.1109/ICOEI51242.2021.9453072.1713-1721). IEEE.
53. Kasapbaşi A, Elbushra AE, Omar AH, Yilmaz A (2022) DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. Comput Methods Programs Biomed Update 2:100048. https://doi.org/10.1016/j.cmpbup.2021.100048.2022
54. GitHub (2018) GitHub - rrupeshh/Simple-Sign-Language-Detector: Simple Sign Language Detector. [online] Available at: <https://github.com/rrupeshh/Simple-Sign-Language-Detector> [Accessed 24 May 2022].
55. GitHub (2018) GitHub - EvilPort2/Sign-Language: A very simple CNN project.. [online] Available at: <https://github.com/EvilPort2/Sign-Language> [Accessed 24 May 2022].
56. GitHub (2020) GitHub - Ahmed-KASAPBASI/Success_Team_ASL: Project of Deep Learning (ASL). [online] Available at: <https://github.com/Ahmed-KASAPBASI/Success_Team_ASL> [Accessed 24 May 2022].
57. Imran A, Razzaq A, Baig IA, Hussain A, Shahid S, Rehman TU (2021) Dataset of Pakistan Sign Language and Automatic Recognition of Hand Configuration of Urdu Alphabet through Machine Learning. Data in Brief 36:107021. https://doi.org/10.1016/j.dib.2021.107021
58. Zhang S, Zhang Q (2021) Sign language recognition based on global-local attention. J Vis Commun Image Represent 80:103280. https://doi.org/10.1016/j.jvcir.2021.103280
59. Sharma A, Mittal A, Singh S, Awatramani V (2020) Hand Gesture Recognition using Image Processing and Feature Extraction Techniques. Procedia Comput Sci 173:181–190. https://doi.org/10.1016/j.procs.2020.06.022
60. Bantupalli K, Xie Y (2018) American sign language recognition using deep learning and computer vision. In: 2018 IEEE International Conference on Big Data (Big Data). Seattle, WA, USA, pp 4896–4899. https://doi.org/10.1109/BigData.2018.8622141
61. Reddy Karna SN, Kode JS, Nadipalli S, Yadav S (2021) American Sign Language Static Gesture Recognition using Deep Learning and Computer Vision. 2nd International Conference on Smart Electronics and Communication (ICOSEC), 1432–1437, https://doi.org/10.1109/ICOSEC51865.2021.9591845
62. Elsayed N, ElSayed Z, Maida AS (2022) Vision-Based American Sign Language Classification Approach via Deep Learning. arXiv preprint arXiv:2204.04235.
63. Rakshit A. Smart learners: Choosing What to Learn Using Bimodal Distribution Removal. Availabale at: http://users.cecs.anu.edu.au/~Tom.Gedeon/conf/ABCs2018/paper/ABCs2018_paper_10.pdf
64. Kaggle.com (2018) ASL Alphabet. [online] Available at: <https://www.kaggle.com/grassknoted/asl-alphabet> [Accessed 24 May 2022].

65. Bilgin M, Mutludoğan K (2019) American Sign Language Character Recognition with Capsule Networks. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 1–6, https://doi.org/10.1109/ISMSIT.2019.8932829.
66. Makhashen GMB, Luqman HA, El-Alfy EM (2019) Using Gabor filter bank with downsampling and SVM for visual sign language alphabet recognition. 2nd Smart Cities Symposium (SCS 2019), 1–6, https://doi.org/10.1049/cp.2019.0188.
67. Rathi D (2018) Optimization of transfer learning for sign language recognition targeting mobile platform. arXiv preprint arXiv:1805.06618.
68 Fregoso J, Gonzalez CI, Martinez GE (2021) Optimization of Convolutional Neural Networks Architectures Using PSO for Sign Language Recognition. Axioms 10(3):139. https://doi.org/10.3390/axioms10030139
69. Hasan MM, Srizon AY, Sayeed A, Hasan MAM (2020) Classification of Sign Language Characters by Applying a Deep Convolutional Neural Network. 2020 2nd International Conference on Advanced Information and Communication Technology (ICAICT), pp. 434–438, https://doi.org/10.1109/ICAICT51780.2020.9333456.
70. Beniwal R, Nag B, Saraswat A, Gulati P (2022) Static Hand Sign Recognition Using Wavelet Transform and Convolutional Neural Network. In: Luhach AK, Poonia RC, Gao XZ, Singh Jat D (eds) Second International Conference on Sustainable Technologies for Computational Intelligence. Advances in Intelligent Systems and Computing, vol 1235. Springer, Singapore. https://doi.org/10.1007/978-981-16-4641-6_13
71. Kaggle.com (2020) Sign Language MNIST | Kaggle. [online] Available at: <https://www.kaggle.com/c/sign-language-mnist> [Accessed 24 May 2022].
72. En.wikipedia.org (2022) Sign language - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Sign_language> [Accessed 24 May 2022].
73. TensorFlow (n.d.) Transfer learning and fine-tuning | TensorFlow Core. [online] Available at: <https://www.tensorflow.org/tutorials/images/transfer_learning> [Accessed 24 May 2022].
74. Tan M, Le Q (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In International conference on machine learning. PMLR, pp 6105–6114
75. Scince Direct (n.d.) Deep convolutional neural network–based image classification. [online] Available at:<https://www.sciencedirect.com/topics/engineering/convolutional-neural-network> [Accessed 24 May 2022].
76. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient- based learning applied to document recognition. Proc IEEE 86(11):2278–2324
77. OpenCV (n.d.) About - OpenCV. [online] Available at: <https://opencv.org/about/> [Accessed 24 May 2022].
78. Chollet F others (2015) Keras. Available at: https://github.com/fchollet/keras. Software available from https://keras.io
79. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S (2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467. Software available from https://tensorflow.org.
80 Wadhawan A, Kumar P (2019) Sign language recognition systems: a decade systematic literature review. Arch Comp Methods Eng 28:785–813
81. Google Developers (2020) Multi-Class Neural Networks: Softmax | Machine Learning Crash Course | Google Developers. [online] Available at: <https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax> [Accessed 24 May 2022].
82. Agarwal V (2020) Complete Architectural Details of all EfficientNet Models. [online] Medium. Available at: <https://towardsdatascience.com/complete-architectural-details-of-all-efficientnet-models-5fd5b736142> [Accessed 24 May 2022].
83. Sarkar, D.: a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a @ towardsdatascience.com, https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a