Measurement: Sensors 23 (2022) 100385

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Survey on sign language recognition in context of vision-based and deep learning

S. Subburaj [*], S. Murugavalli

Department of CSE, Panimalar Engineering College, Anna University, Chennai, India

ARTICLE INFO

Keywords:
SLR
Sign language
Recognition
Computer vision
Neural networks
Deep learning
HMM
CNN

ABSTRACT

Every day we see many people with disabilities like the deaf, the dumb and the blind, etc. Sign language is one of the communication tools for the hard-of-hearing people community and common people community. But, normal people find it hard to understand the sign language and gestures of the deaf and dumb. Many tools can be used to translate the sign language created by the disabled into a form that normal people can understand. The studies are based on various image acquisition, preprocessing, hand gesture segmentation, extraction of features, and classification methods. This paper aims to research and examine the methods employed within the SLR systems, and the classification methods used, and to propose the most promising technique for future research. Due to the latest advancement in classification methods, a few of the currently proposed works specifically contribute to classification methods, together with hybrid techniques and deep learning. This paper specializes in the classification strategies utilized in earlier Sign Language Recognition. Based on our review, HMM-based techniques were explored significantly in previous studies, which include modifications. Deep learning consisting of convolutional neural networks has become popular over the past five years.

1. Introduction

This paper addresses the various algorithms and techniques that can be used to understand the sign language and hand gestures of hard-of-hearing people. The hand gesture recognition system is considered a method for more intuitive and efficient interaction with humans and computers. The program range includes virtual prototyping, an examination of sign language, and medical training.

Sign language is one of the communication tools for hard-of-hearing people communities and common people communities [6]. Currently, research work has mainly focused on the identification of static sign language signs from images or video sequences captured under controlled conditions. Signers in this category are required to wear a glove sensor or a shaded glove. The task will be standardized by wearing gloves during the segmentation process. The disadvantage of this approach is that during the device service, the participant will wear the sensor components together with the gloves.

Methods Based on Vision: Computer-based vision strategies are noninvasive and based on how people perceive their environmental knowledge. Although developing a vision-based interface for general use is challenging, designing such an interface for a controlled environment is still achievable. Feature selection is crucial to the recognition of gestures since hand gestures are very unique in shape variation, textures, and motion. For static hand recognition, it is easy to recognize hand posture by extracting some features such as finger directions, fingertips, skin colour, and hand contours. Such features are not always available and reliable due to lighting conditions and the background of the image. There are also many other non-geometric features, such as the silhouette, colour, and textures, that are inadequate in recognition. Since it is not easy to define features clearly, the entire frame or transformed image is taken as the input, and features are chosen automatically and implicitly by the recognizer. This paper aims to review and evaluate the approaches used in previous studies. It also aims to recommend the best method to investigate for future research. Majid and Zain (2013) were studied the development of Sign Language Recognition devices for various sign languages. They only studied the best 32 related publications up to 2012.

[*] Corresponding author.
E-mail address: subburajs87@gmail.com (S. Subburaj).

https://doi.org/10.1016/j.measen.2022.100385
Received 31 May 2022; Received in revised form 23 June 2022; Accepted 24 June 2022
Available online 15 July 2022
2665-9174/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

2. A review of the sign language recognition system

2.1. Sign language

The word sign language is similar to the language phrase, many of both are spread around different world territories [7]. Similar to language, sign language evolves over a long period of period sign language grammar and vocabulary, so it is considered a legitimate language. Because no perception of hearing is needed to understand sign language and no voice is needed to produce sign language, it is the common language among the deaf. Sign languages [84] are usually constructed by using simultaneous compilation associated with hand shapes, orientations, and moves of the hands, palms, or body, along with facial expressions to fluidly explicit a speaker's thoughts.

2.2. Sign capturing methods

The signs must be captured to provide input for the sign language recognition system. To capture images of hand gestures through a Microsoft Kinect camera which handles single-hand signs, double-hand signs, and finger-spelling [4,12,13], Microsoft Kinect sensors to capture multimodal data [6,72–74], Microsoft Kinect (RGB-D) sensor handled by the Nui Capture Analyze application [7], front cameras and mobile cameras [5,8,11], Sony video cameras [9], and Cannon 600 D camera RGB videos [10] are used. Microsoft Kinect was initially designed as a Gaming console peripheral device. The three sensors, that is, RGB, audio, and depth allow movements to be detected and user faces/speeches to be recognized. Microsoft Kinect sensors use a variety of useful computer vision applications, including gesture recognition, motion recognition, robotics, and virtual reality.

3. Sign language recognition techniques

There are again two different approaches to vision-based sign language recognition: appearance-based and deep learning-based.

Appearance-based approaches are modeled by a collection of 2D intensity images. In turn, gestures are modeled as a sequence of views. Appearance-based approaches attempt to infer the pose of the palm and the joint angles [18,19]. Appearance-based frameworks use images or videos as data sources. They straightforwardly interpret these videos/images. They do not utilize a spatial representation of the body. The parameters are usually derived straightforwardly from the images or videos using a template data repository.

Traditional Machine Learning-based Approaches: A famous AI approach is to regard a motion as the output of a stochastic process. Of this category of approaches, CNN has received the most consideration in the literature for classifying gestures.

4. Appearance-based sign language recognition system

A simple block diagram of the appearance-based SLR system is shown in Fig. 1.

4.1. Image acquisition

The important element used in the sign language recognition method (SLR) as the input method is the camera. The input data for the SLR are in the form of a moving image that can easily be recorded by a camera. Nevertheless, some researchers use normal cameras to capture images [1,2,4–7,11],] [17,22,49,53,54]. Some researchers claim that they are using cameras and no gloves to reduce the challenge of using sensor-based gloves. Cameras usually support many video formats, so we need to define the default format and the format that we want to utilize by using Digitizer Configuration Format (DCF) file. Some researchers have used higher-quality cameras because the image of the web camera is blurred. A camera was used to capture 30 frames per second of real-time video and then analyzed frame by frame for dynamic gestures. The system uses a skin filter to extract the skin region and is then converted into HSV color space for each frame to an image.

There is also another device named Microsoft Kinect [14,42,47,48,51,75] that is used to capture images. Nowadays, because of its feature, Kinect is commonly used by researchers. Kinect can simultaneously have colour video streams and depth video streams. Background segmentation can be easily done with depth data and can be accomplished with Kinect by using signal language recognition.

Many researchers have used predefined datasets from the American Sign Language Image Dataset (ASLID) [67],ASL Gesture Dataset 2012 [61], ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) – 2010 [83], ChaLearn Looking at People 2014 (CLAP14) [84], RWTH-Phoenix-Weather 12 [57,85], RWTH-Phoenix-Weather Multi-signer 2014 [85], SIGNUM and ArSL databases [44,86], ASLU [9][87], Myo Armband [45], and RWTH-BOSTON-50 [49] [88]. Few researchers create their own datas for their training of data. Because of the lack of availability of sign language datasets in particular region languages. Researchers record the data from the signer to create a dataset. ASL signs represent letters of the English alphabet [1] shown in Fig. 2 and ISL two-hand signs are shown in Fig. 3.

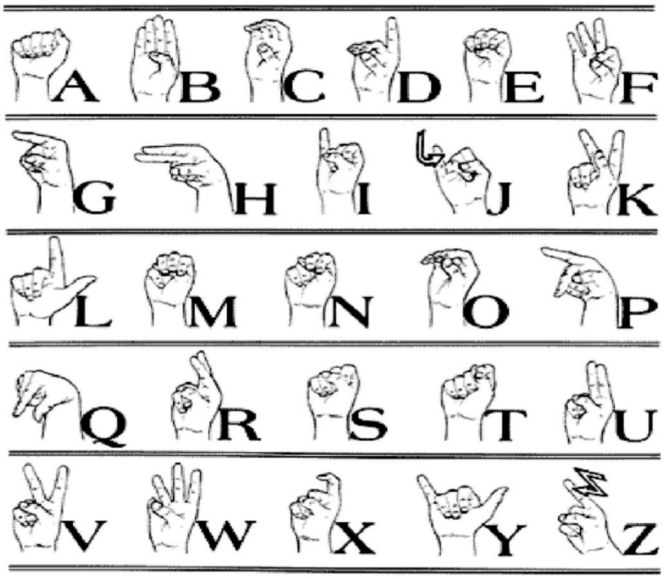

Fig. 2. ASL one hand signs.

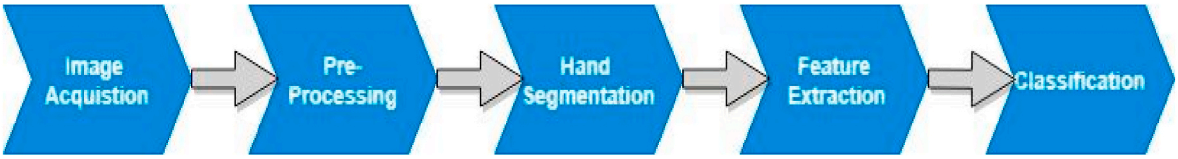

Fig. 1. Appearance-based SLR system.

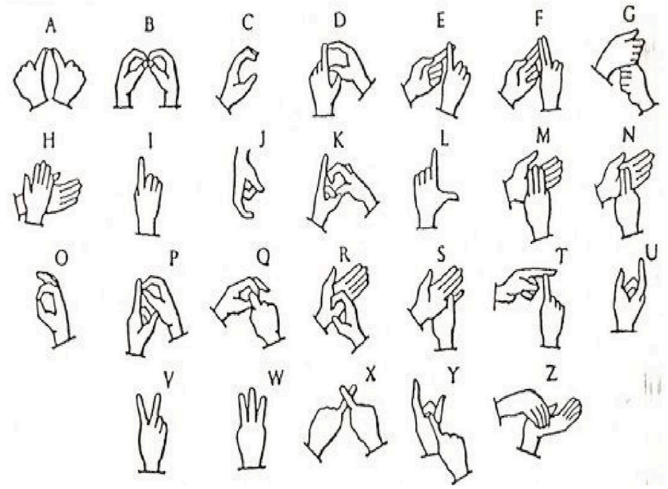

Fig. 3. ISL Two hand signs.

4.2. Preprocessing and segmentatio

The image pre-processing step improves the system input to modify the image and videos. Median and Gaussian filters are some of the most frequently used methods to reduce noise in input images or videos. In research [7,9,48,49,76] median filtering is exclusively used for the image pre-processing stage and morphological operations [20] are also broadly used to remove unwanted information from the input. For instance, Badhe et al. [1] and Krishnaveni et al. [11] threshold the input image into binary and then K-means clustering with morphological operations in the pre-processing stage to remove noise. An adaptive histogram [6] is used to improve the contrast of input images acquired in different environments.

The segmentation approach can be contextual or non-contextual. Contextual segmentation considers the spatial connection between highlights, for example, edge recognition strategies, while a non-context-oriented division does not consider spatial relationships but bunches pixels dependent on global attributes. Skin detection also applies hand movement tracking with skin detection to produce more specific end results. Similarly to skin detection, coloured gloves are used to provide distinct characteristics to the hands, thus aiding in hand segmentation.

Skin colour segmentation is processed in the RGB model, HSV model, HIS model, and YCbCr colour models [52], while colour segmentation follows difficulties because we may face sensitivity to illumination, cameras, and skin tone. The HSV colour model is famous because the hue of the hand used differentiates palm from arm easily. In research [9], the Palms and faces of people were segmented according to HSV and YCbCr colour models. Ahmed et al. [44] used the RGB colour model to carry out hand skin colour segmentation. Match and compare with the RGB colour model used to find skin colour in a given image or video. According to research [22,54], the YCbCr colour model was found to be useful for colour segmentation under various light conditions.

The discrete cosine transform [6], the Viola Jones algorithm [7] and The Gaussian distribution low pass filter (grayscale) gives powerful colour segments to the face and hand in the grayscale model. In research [42] proposed the use of K-means clustering in YCbCr colour space to isolate the frontal area from the background in an image. Badhe et al. introduced a hand tracking movement in Ref. [1] to track hands in the video. A Canny edge detector is achieved by erosion and dilation. The edge traversal algorithm segments the hand motion from the background in the video.

Hand segmentation is a technique to segregate hands and different features from the rest of the image in vision-based systems. Rao et al. [6] employed the DCT & Viola Jones algorithm to do the frame pruning and utilized coloured gloves to aid adaptive histogram hand-head segmentation. Many hand segmentation techniques have been proposed in computer vision. The Canny edge detector is used to detect the palm edges of an image. The Canny edge detector is known for its ideal performance in identifying edges [7].

The other strategy for hand segmentation is Harris corner detection, which is used to find articulation points and motion of hands [14]. Boulares et al. [47] discover hand segmentation with 2D hand signature analysis using motion data matrices. Morphological operations [20] extract the elements of an image, which are helpful in the representation and description of region shape, i.e., skeletons, boundaries, and convex hulls.

4.3. Feature extraction

Feature extraction is the process of extracting multiple features from an image. The features are image background, translation of image, scaling, shaping, rotation, angle, and coordinates. To extract the external boundary of objects in images, Fourier descriptors [1,7,9,42,45] are used. The sequence of coordinates forms boundaries to identify objects in an image. The Horn-Schunck optical flow algorithm [5] extracts tracking points for both arms in every frame. In Almeida et al. (2014) [14], the Speeded Up Robust Features (SURF) algorithm has been used as a feature extraction strategy in prior research. SURF is a patented descriptor for finding local features in a video.

In the Hough transform [14], the elements are arranged in into pairs (q, h) since we utilize polar directions to identify lines. It is used to find two-hand communication features for the recognition of SL. HOG [17], which is broadly utilized in the segmentation stage. Haar classifiers have been used for object recognition and used for initial real-time face detectors [9]. Local binary patterns (LBPs) [44,48], find the surface and shape in grayscale images. LBP is, by all accounts, good with different facial expressions and rotation of an image. Therefore, it is reasonable for extraction in gesture-based recognition. Other feature extraction methods are tracked particle filters [49], 121 points used as basic descriptors [51], Zernike moments for keyframe extraction [52], and the distance algorithm [20], which are used to extract features for classification in the final step.

4.4. Classification

Classification is the final stage and an essential level in the popularity of gestures. Words or sentences in sign language are made from continuous gestures, with modifications over time. Consequently, a reputation approach must be capable of handling sequential information. A few problems occur when the device handles noisy facts and uncontrolled surroundings. The method of popularity is to pick out the model from the set of fashions that could properly represent the phrase series. There are two varieties of gesture popularity processes. A few researchers have used the extracted functions for gesture recognition, which include template matching, and some have used machine learning classifiers consisting of Hidden Markov models (HMM).

4.4.1. Machine learning classifiers

We chose the Hidden Markov model (HMM) [44,53], which has the highest probability of generating the given collection and the sign with the highest probability. The method of popularity is to pick out the model from the set of fashions that could properly represent the phrase series. Support vector machines (SVMs) [14,17,45,47,48,52] are used to classify signs based on these features and linguistic elements. The SVM classifier is a multiclass classifier that searches for an ideal hyperplane as a decision function. Once trained on images containing some particular gestures, the SVM classifier can make decisions regarding the sign.

Random forest (RF) [51] is a type of machine learning that can classify regression problems. Certain features of the object are chosen as the benchmark when a new classification begins. A final prediction is obtained by aggregating all constructed trees through majority voting. Other machine learning-based classifiers are backpropagation training algorithms [5], AdaBoost multiclass [6], Sugeno-type fuzzy inference systems [7], ANNs [20], and MPCNNs [22].

4.4.2. Template matching

An appropriate symbolic similarity measure is studied to establish matching among test and reference signs, and a simple nearest neighbor classifier [54] is used to recognize an obscure sign as viewed as one of the recognized signs by indicating a preferred level of threshold. Euclidean Distance [1,4,14,42,49] Every gesture image in the testing dataset is compared against each gesture in the training dataset by using Euclidean distance. The gesture with a minimum distance is considered a match.

5. Traditional machine learning-based approaches

5.1. Data acquisition/image acquisition

The important element used in signal language recognition (SLR) as the input method is the camera. The input data for the SLR is in the form of a moving image that can easily be recorded by a camera. Nevertheless, some researchers use simple cameras to capture images. Some researchers still use simple cameras [8,10,12,15,23,25,31,36,55] to capture images. There is also another device named Microsoft Kinect, which is used to capture images. Nowadays, Kinect is widely used by researchers because of its features. Kinect can offer colour video streams and depth video streams concurrently. With depth data, background segmentation can be carried out easily [13,26,29,34,43,50]. used Kinect for sign language recognition.

5.2. Datasets

Most researchers create their own datasets for the training of their data. Because of the non-availability of sign language datasets in particular regions, researchers record the data from the signer to create a dataset. Researchers prepare their own sign language datasets because they usually do not have enough datasets to use for research. Numerous researchers have used predefined datasets from the American Sign Language like Image Dataset (ASLID) [27], ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) – 2010 [33,83], ChaLearn Looking at People 2014 (CLAP14) [84], RWTH-Phoenix-Weather Multisigner 2014T [16,18,37–39,41,85], SIGNUM [37] and The ArSL database [86], the Massey University dataset [19] [35] [46] [89], and the AND VIVA challenge dataset [21,82].

The latest update on the pre-processing method and experiments using active sensors was conducted in Ref. [3]. They suggested a feature extraction method using the data obtained by using a Leap Motion Controller (LMC) [24,28]. An LMC is a device that can identify hand movements at 200 fps and gives an identity whenever it detects hand movements. The particular LMC API can directly map the detected data to hand fingertips and movements. However, LMC is not perfect and is still developing. It has some difficulties in implementing the API when hands are flipped over. The Leap Motion controller [4] is a smaller and more commercialised sensor for hand and finger motions in a 3D space of approximately eight cubits above the device. The sensor reports data such as the position and speed of the hand and fingers, primarily based on the sensor's coordinate system. Data is transferred to a computer using a USB connection.

5.3. Preprocessing/pre-trained model

5.3.1. Pre-trained model

A pre-trained model is a model that has been trained on a large benchmark dataset to solve a problem similar to the one that we want to solve. Due to the computational cost of training such models, it is common practise to import and use models from published literature.

AlexNet [8,15,16,19,27,30] Researchers are also focused on mature networks that can be transmitted in transfer learning. From that, AlexNet is one of the lights that ignited the deep learning explosion, which was developed by LeNet [87]. Even though AlexNet in terms of structure, there are no notable dissimilarities between LeNet and AlexNet. There are a few differences: Alex Net is attached with ReLU activations at the end of the convolutional layer, dropout, and data augmentation to avoid overfitting while training, convolutions, max pooling, and overlapping.

GoogLeNet [10,31,37,38,41,46,66] is a neural network with depth in both the vertical and horizontal directions. In neural networks, a horizontal direction having width is additionally referred to as an "inception structure," which makes use of more than one filter of a unique size and, in the end, combines their results. The "inception structure" was suggested to assemble a primary neuron structure to construct a sparse and high-performance computing neural network structure. The dimension of the feature map was reduced by inserting a 1 × 1 convolution layer in front of the 3 × 3 and 5 × 5 convolution layers, and max-pooling was introduced before the 1 × 1 convolution. In this way, the basic shape of Inception V1 [59], is formed. Next, optimizations and enhancements were run on the foundation of inception V1, resulting in V2, V3, and V4 neural network structures.

VGG16 [33,65]The VGG network architecture was presented by Simonyan and Zisserman in their 2014 paper [77]. The VGG family of networks is described by using only 3 × 3 convolutional layers stacked on top of one another in increasing depth. Diminishing the volume size is dealt with by max pooling. Two fully-connected layers, each with 4096 nodes, are then trailed by a softmax classifier. VGG-16 networks were considered very deep, although we now have the ResNet model structure [62], which can be successfully trained at depths of 50–200 for ImageNet and over 1000 for CIFAR-10. Unfortunately, there are two major disadvantages to VGG:

1. It is very slow while training.
2. The network weights are quite large.

Because of its depth of layers and number of trailing fully-connected nodes, the serialised weight files for the VGG16 network are 533 MB in size.

The ResNet50 [35,39] architecture has proven to be a seminal work in the deep learning literature, demonstrating that extremely deep networks can be trained using standard SGD (and a reasonable initialization function) by using residual modules. ResNet is extensively deeper than both VGG19 and VGG16, and the model size is entirely more modest due to the use of global average pooling rather than fully connected layers, which reduces the size of the model down to 102 MB for ResNet50. Other pre-trained models are Squeeze net [3], Mobile Net [31,60], C3D model [40,70], and Creative Senz3D camera [32], which are used to copy the learned features (weights) to new model neural networks for learning.

5.3.2. Preprocessing

The image pre-processing stage is done to modify the photo or video inputs to improve the standard overall performance of the system. Median and Gaussian filters are some of the most regularly used techniques to reduce noise in images or videos obtained. The pre-processing method for extracting features from the training image and these features are stored in neural networks to classify images in testing images.

Pre-processing methods are: Haar feature classifier [36], Nearest neighbor interpolation [21], Image background subtraction [55], Median filtering [50], Bandpass filter [43], Gabor filter [34], Savitzky-Golay filter [24,28], RGB to HSV colour space [12] and HOG [29].

5.4. Neural network model

5.4.1. CNN [3,15,19,20,23,31,35,36,39,43,46,55]

A convolutional neural network (CNN) is perceived as the most important deep learning neural network model to perform with regards to recognising and classifying images. It uses multilayer superposition to extract low-level features into relevant features, resulting in a hierarchical structure similar to simulations of human brain activity [3,15,19,20,23,35,36,39,43,46,55]. The procedure of lengthy manual feature extraction can be prevented because the recent features are passed from the past layer. CNN combines both feature learning and classification. A convolutional neural network is made up of many layers in general. In the convolution layer, a convolution operation is used to extract features from an input layer or a prior layer. The pooling layer can constantly shrink the data's space size, reducing the number of features and computations. In the CNN, the fully connected layer serves as a "classifier." A simple CNN diagram is represented in Fig. 4.

CNN automatically learns the values from these layers. In the context of image classification, our CNN may learn to identify edges, identify shapes, and help boundaries identify higher-level features such as face structures, respectively with the first, second, and highest layers of the network by applying convolution filters, nonlinear activation functions, pooling, and back propagation.

5.4.2. 3D CNN [21,29,40,50]

A CNN architecture can be built by multiple layers of convolution and pooling in an alternating fashion. 2D CNNs are applied to image datasets to classify them and extract spatial features. Anyway, for SLR in videos, both spatial and temporal information are required to be captured. 3D CNN carries out convolution in videos to extract both spatial and temporal information. Our version learns and extracts each spatial and temporal capability through the performance of 3D convolutions.

5.4.3. CNN-RNN(LSTM) [10,12,16,18,24,28,38,41]

A CNN architecture can be built to gather spatial features from the video for SLR and then extract temporal features from the video by utilizing an LSTM (long-term memory) and an RNN (recurrent neural network) model. LSTM identifies gesture classes using the sequence information in SLR video.

5.4.4. Deep-CNN [27,32,33]

Deep convolutional neural networks (DCNNs) to learn conditional probabilities for the presence of components and their spatial relationships within image patches. Video sequences in which the temporal structure provides very helpful data where this is missing or in ways less obvious in static images. Other neural network models work similar to CNN, like faster R–CNN [8,37], CNN-dynamic Bayesian network (DBN) [13], stream CNN [26], and attention-based RNN [30], which are used to extract spatial features from videos and long sequences of the pose.

5.5. Loss function

A loss function quantifies how well our predicted class labels accept as true with our ground-truth labels. The higher the degree of agreement among those sets of labels is, the decrease our loss. Preference for loss function is directly related to the activation function used within the output layer of your neural network. Those two design elements are related. Categorical cross entropy [3,10,15,16,18,26,31,35,43,55]. Categorical cross-entropy is a loss function that is utilized in multiclass classification tasks. These are tasks in which an instance can only belong to at least one out of many feasible classes, and the model needs to determine which one. Every predicted possibility is in comparison to the actual class output value (0 or 1), and a score is calculated that penalizes the probability primarily based on the distance from the predicted value.

Negative log-likelihood [12,21,36,40,50] loss function while in training, we need to find the minimum loss value in given parameters. Here we interpret loss as "unhappiness "in the network. We need to build a network model happy.

5.6. Optimizer

An optimizer is used to reduce the loss function by updating parameter values in the neural network model. The loss function is the guide for the optimizer to choose the right or incorrect direction. Gradient descent is an iterative algorithm that starts at some random point and travels to the slope of the destination to the lowest point in the function.

Stochastic gradient descent [3,8,21,23,25,26,28,33,35,38–40,50,55] (SGD), a simple alteration to standard gradient descent algorithm programming, calculates the gradient and updates the weight matrix value. W on mini-batches of training data instead of the complete training set. SGD is arguably the most essential algorithm for training deep neural networks. That is a modified version of the GD method, wherein the model parameters are up to date on each iteration. This means that after each training sample, the loss function is examined and the version is updated. These frequent updates bring about convergence to the minima in less time, but it comes at the cost of multiplied variance that may make the model overshoot the required position.

ADAM [10,15,16,18,34,38,39] Adaptive moment estimation (Adam) [14] is another good method that computes adaptive learning rates for every parameter. Adam additionally maintains an exponentially decaying average of past gradients similar to momentum. While momentum can be seen as a ball strolling down a slope, Adam behaves like a heavy ball with friction, which consequently prefers flat minima in the error surface.

ADADELTA [12,22,76] is an extension of Adagrad that seeks to reduce its competitive, monotonically reducing learning rate. Instead of collecting all past squared gradients, Adadelta restricts the window of accumulated past gradients to a few constant lengths.

ADAGRAD [8,36] is an algorithm for gradient-based optimization that does just this: It adapts the learning rate to the parameters,

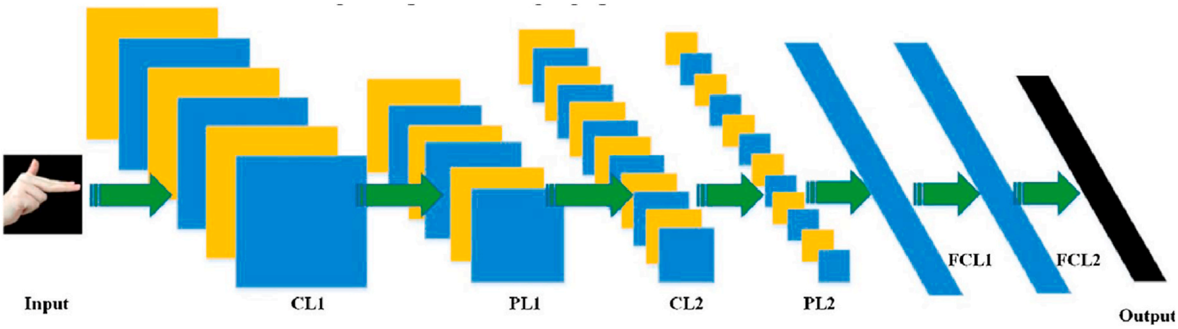

Fig. 4. Convolutional neural networks.

performing larger updates for occasional and smaller updates for recurrent parameters. For that reason, it is properly perfect for handling sparse information. Adagrad significantly advanced the robustness of SGD and used it for training at Google large-scale neural nets, which – amongst different things – learned to understand cats in YouTube videos. The Adagrad optimizer attempts to provide adaptiveness with the aid of decaying the learning rate in proportion to the updated history of the gradients. In this method, when there are large updates, the history component is gathered, and consequently, it reduces the learning rate and vice versa. One drawback of this technique is that the learning rate declines aggressively and after a while, it approaches zero. Adaptive moment estimation combines the power of momentum-based GD and RMSProp (root-mean-square prop).

In Adam optimizers, the power of momentum GD to preserve the records of up to date and the adaptive learning rate supplied by way of RMSProp creates Adam optimizer as an effective method. Adaptive optimization strategies together with Adam or RMSprop perform well within the initial part of training, but they have been observed to generalize poorly at later stages in comparison to stochastic gradient descent.

5.7. *Classification*

Classification finds a function to determine the category to which the input data belongs. It can be two-category or multi-category. Factors including the classification construction method, the properties of the data to be classified, and the number of training samples all influence classification accuracy.

Softmax function [3,8,10,12,13,15,18,21,23,25–27,30,31,33,35–39,41,43,46,50,55,78], **the** Softmax classifier finds classes based on final output probabilities in an efficient manner, and it implies multi-classification networks. In Softmax, every class is assigned decimal probabilities in the multiclass problem, and probabilities add up to 1.0. The categorical cross-entropy loss function is mostly used with the softmax activation function. The output of the model is most effective because of its logarithmic output value. The softmax activation rescales the model output so that it has the right properties. Because of this, it is common to append a softmax function because of the final layer of the neural network.

ReLU [23,25,32,34] are also known as "Ramp functions" due to how they look when plotted. Notice how the function is 0 for negative inputs but then linearly increases for positive values. Although the ReLU is not always saturable, it is tremendously computationally efficient. ReLU has no vanishing gradient problem compared to nonlinear functions because the gradient is constant within the nonnegative region. Learning and optimising the ReLU function is significantly easier.

To prevent overfitting, the ReLU function reduces several neurons to zero in the output, making the network sparse.

$$f\ (x) = \max(\text{zero};\ x)$$

Sigmoid [28], When a sigmoid function is a neuron's activation function, the output of this unit will fall between 0 and 1. When the sigmoid function is introduced as a nonlinear activation function, the output would be the sum of the inputs to the nonlinear function.

In the early days, the sigmoid and tanh functions were used as nonlinear functions more frequently, which limited the output.

The sigmoid function is a better preference for learning than the simple step function because it:

1. is continuous and differentiable everywhere.
2. is symmetric across the y-axis.
3. Asymptotically, its saturation values increase

The primary benefit here is that the smoothness of the sigmoid function makes it easier to devise learning algorithms.

Different classifiers like Support Vector Machines(SVM) [19], Couple Hidden Markov Models (CHMMs) [13], and MultiLayer Perceptron (MLP) classifiers [29], perform multiclass classification on the final layer of neural networks.

6. Discussion

This section gives an outline of previous surveys done on gesture and sign language recognition work, as well as the techniques applied in various studies. Information, including techniques applied and performance, is presented and tabulated in this section. A summary of the techniques used in appearance-based SLR can be seen in Table 1, while a summary of traditional machine learning-based approaches can be found in Table 2.

Table 1 lists the techniques used and is categorised by classification, feature extraction, segmentation, pre-processing, and image capturing. Table 2 lists the techniques used and is categorised by the image capturing, pre-processing and pre-trained model, neural network model, loss function, optimizer, classification, and accuracy. The accuracy column shows the highest accuracy achieved by the proposed method.

Kinect cameras and normal cameras are most significantly used for data acquisition. Data acquisition is an important step in feeding input to the SLR system. Pre-processing is done after data acquisition and is required to improve the accuracy and collect more information from given data. In pre-processing, median and Gaussian filters are used to remove noise from given data. Before segmentation, images are downsized to reduce computational time. Most significantly, skin colour segmentation is used in segmentation. The HSV, YCbCr, and RGB colour spaces are efficiently differentiated from the background colour and skin colour. The research showed that skin colour segmentation with other features, such as threshold and edge identification, improves the segmentation result.

The classification of gestures is the final step in the appearance-based strategy, and it extracts features from images for easy identification. ANN and SVM are the most commonly used classification algorithms. SVM provides high performance when evaluation is done by researchers. The hidden Markov model (HMM) is a general method to recognize sign language because it is employed in statistical methods to get spatio-temporal information. In review with existing data, most models use sensors to collect data from the environment. While HMMs and SVMs are used as classification methods, with the improvement of data source availability, neural networks are broadly used in vision-based approaches to images and videos.

The neural network model is another important processing method for sign language recognition. The CNN process image convolves through convolution layers, pooling, activation functions, and fully connected layers to perceive Sign Language. Applied 3D-CNNs process video streams to acquire features of temporal and spatial correlation and to get motion details through depth variation in frames. LSTM-based methods can get simulation temporal sequence information from sign language videos [79]. An advancement in LSTM called BLSTM-3D ResNet [80] can localise palms and hands from video sequences and separate spatial information. HMM and SVM are important classification methods in appearance-based systems. Recently, convolutional neural networks have been vital in vision-based sign language recognition research.

Pre-trained models resemble magic, and we can simply download the models and start utilizing them, even without any data input or training. In recent years, the promising idea of pre-trained models has attracted widespread consideration by researchers. It also reduces the cost of datasets and training. Many researchers have used gesture-based vgg16, Google net, and AlexNet pre-trained models to reduce the cost of training. Loss functions like cross-entropy are used to reduce loss errors while training datasets. The most commonly used loss function is the cross-entropy multiclass category. The loss function is required to work with the optimizer, SGD, and ADAM optimizers widely used in training

Table 1
Appearance based approach.

Author/year	Image Acquisition/ dataset	Pre-Processing	Hand segmentation	Feature extraction	Classification
Badhe et al. (2015) [1]	Camera	Threshold image	Canny Edge detection (Skin colour with yCbCr)	2D FFT Fourier Descriptor	Euclidean Distance -Template matching
Nanivadekar et al. (2014) [2]	Digital camera video recorder	Video converted into frames	Gaussian (skin colour detection)	–	–
Nandy et al. (2010) [4]	Normal camera	–	HMM	Orientaion histogram	Euclidean Distance
P. V. V. Kishore and et all (2013) [5]	HD Sony camcorder		Active Contours (AC) extract shape features	Horn Schunck optical flow (HSOF)	backpropagation training algorithm
Rao et al. (2018) [6]	Selfie mode video	DCT& Viola jones algorithm (frame pruning)	Adaptive histogram Hand-head segmentation	Feature matrix with n features	Ada boost mulilabel multiclass
Kishore et al. (2018) [7]	Sony Cybershot H7 digital camcorder	Gaussion low pass filter (gray scale)	Fusing Discrete wavelet transform & canny edge detector	Elliptical Fourier descriptor	Sugeno type fuzzy inference system
Shivansankra et al. (2018) [9]	ASLU	RGB to HSV, HSV to yCbCr colour space	Threshold image Binary erosion & dilation median filter	Central mass of region	Roundness values and number of peaks
Krishnaveni et al. (2012) [11]	Camera	Binary image	Mean intensity area, perimeter	Discriminative classifier	KNN,MLP& SVM
Almeida et al. (2014) [14]	RGB –D sensors and Kinect sensor and NUI Capture analyser	Colour into depth frame and skeleton frame	Harris corner detection	Hough Tansform, SURF, Optical Flow	Euclidean Distance & SVM
Eriglen et all (2016) [42]	Microsoft Kinect device	constant threshold and skeleton feature Kinect	K- means clustering algorithm	Fourier descriptors	Euclidean distance
Ahmed et all (2014) [44]	ArSL database	Convert RGB to YCbCr color space	Gaussian Mixture Model (GMM)	LBP and PCA	Hidden Markov Models (HMM
Savur et al. (2016) [45]	Myo Armband	zero mean and bandpass filters	SEMG signal	Principal Component Analysis (PCA)	SVM and bagged tree.
Boulares et al. (2012) [47]	Microsoft Kinect sensor	motion data matrices	2D hand signature analysis	3D non linear regression	support vector machine (SVM)
RAGHUVEERA et al. (2020) [48]	Microsoft Kinect sensor	Median filtering	K-mean clustering	Local Binary Patterns & Histogram of Oriented Gradients	Support Vector Machine
Lim et al. (2016) [49]	front camera RWTH-BOSTON-50	combination of the median and mode filters	serial particle filter	Tracked Particle filters	minimum of total Riemann distance
GUERRA et al. (2018) [51]	Microsoft Kinect (RGB-D) sensor LIBRAS	xy-coordinates of 121 points	Displacement Ranking	121 points used as basic descriptors	Random Forest (RF)
Athira et al. (2019) [52]	mobile camera	skin colour segmentation	co-articulation elimination phase.	Zernike moments for key frame extraction	Support Vector Machine
Mohandes et al. (2012) [53]	Sony video camera	Gaussian skin model	region-growing technique	morphology of both hands	Hidden Markov Model (HMM)
Chethana Kumara et al. (2016[]54]	Cannon 600 D camera RGB	Hue and HSV color space saturation values	spatial relationships	K-means clustering algorithm	nearest neighbor classification
Lilha et al. (2011) [17]	Still camera Images	Connected Component Labelling (CCL)	logically AND with original image	Histogram of Orientation Gradient (HOG), Histogram of Boundary Description (HBD) and Histogram of Edge Frequency (HOEF)	Support Vector Machine
Akmeliawatil et al. (2007) [20]	Logitech webcam	Sum of Absolute Difference (SAD) algorithm	Morphological opening and closing	Distance Algorithm	ANN
Nagi et al. (2011) [22]	front (CMOS) camera	RGB to YCbCr color space	Single Gaussian Model (SGM)	Erosion and dilation	MPCNN

data.

The deep neural network (DNN) [71] gives better results because it has the ability to self-learn and self-associate, but it requires the largest dataset for training. With recent technological developments in GPUs, it now has the computing power to run applications on large datasets. A new algorithm has been implemented, and improvements to existing algorithms provide better execution in an application. Increasing the computing speed can execute applications in less time with Cloud-based computing and big data applications.

It provides improved access and most researchers' involvement makes deep learning more popular in the research domain. Therefore, more deep learning-based studies have been carried out on neural network technology. Neural network technology mimics the human brain, so every human activity can be implemented through neural network-based applications. Thus, it can reduce tedious human activities and learn a new language like sign language.

7. Benchmark databases

In SLR Research, benchmark databases are available as the standard references for future research. Benchmark databases allow the comparison of model-free and person-independent approaches. Most of the databases for SLR are available as open-source on the internet, i.e., Kaggle [63] and Google datasets. The following databases are referenced by the most popular papers in SLR Research. These include Purdue RVL-SLLL [83], RWTH-Phoenix-Weather 2014, RWTH-BOSTON-50, RWTHBOSTON-104, and RWTH-BOSTON-400. I have found the following datasets to be useful: RWTH-Phoenix-Weather Multisigner 2014T [16,18,37,37–39,41], ATIS Sign Language Corpus [81], American Sign Language Image Dataset (ASLID) [27], ImageNet Large-Scale

Table 2
Deep Learning based Approaches.

Author/year	Data set/image acquisition	Preprocessing/Pre-trained	NN Model	Loss Function	Optimizer	Classifier	Accuracy
Nikhil Kasukurthi and et all(2019) [3]	Intel Real sense P200 Depth camera	SqueezeNet	CNN	Categorical Cross Entropy	Stochastic Gradient Descent	softmax function	83.29%
B. Shi et al. (2018) [8]	You tube Web cam	AlexNet	Faster R–CNN (CNN-LSTM)	–	Stochastic Gradient Descent (SGD)	Softmax Function (CNN) CTC [68]	42%
Ss Shivashankara and et all (2018) [10]	Iphone 6 60 FPS	Inception model (2014)	CNN-RNN	Categorical Cross Entropy	ADAM	softmax function	90%
Su Yang and et all (2017) [12]	Video 10 FPS	RGB to HSV colour space	CNN-LSTM	log-likelihood function	ADADELTA	softmax function	95%
Xiao and et all (2018) [13]	Microsoft Kinect RGB video	Colour and depth frames	CNN-Dynamic Baysian Network (DBN)	–	–	softmax function & (CHMM)	99.40%
Wadhawan and et al. [15] (2020)	WebCamera	AlexNet	CNN	Categorical Cross Entropy	ADAM, ADAGARD, SGD, ADADELTA	multi-classification problem & softmax function	99.72%
Yongsen and et all (2020) [25]	Normal Camera	WiFi Preamble and Channel State information	CNN	–	Stochastic Gradient Descent with Momentum	softmax function & RELU	98.01%
Kishore, P.V.V and et all (2019) [26]	Kinect video Sensor Camera	–	Two Stream CNN	Categorical Cross Entropy	Stochastic Gradient Descent	softmax function	92%
Srujana Gattupalli and et all (2016) [27]	ASLID Dataset	AlexNet, Caffe and chainer	CNN-Deeppose network [58]	–	–	softmax function	–
Avola and et all (2019) [28]	Leap Motion Controller (LMC)	Savitzky-Golay filter [69]	DLSTM(RNN)	Kronecker delta	Stochastic Gradient Descent	Tanh or sigmoid	91.43
Huang and et all (2015) [29]	Microsoft Kinect RGB video	HOG & GMM-HMM	3D CNN	–	–	Multilayer perceptron classifier	94.2
Shi and et all (2019) [30]	ChicagoFSWild data set & Amazon Mechanical Turk	AlexNet	Attention-based RNN	–	–	Softmax Function & CTC	61.2
Camgoz and et all (2018) [16]	RWTH-PHOENIX-Weather 2014T	AlexNet	CNN-RNN-HMM (Sign2Gloss2Text)	Cross Entropy Loss	Adam	CTC	26% (WER)
Camgoz and et all (2017) [18]	RWTHPHOENIX-Weather –2014	BLVC CaffeNet [64]	CNN& Bi- directional LSTM	Cross Entropy Loss	Adam	Softmax Function	80.6%
Kika and et all (2018) [19]	Massey University Dataset	AlexNet	Histogram of oriented gradients & CNN	–	–	Support Vector Machines	HOG-87.69% CNN-86.15%
Fang and et all (2017) [24]	Leap Motion Controller (LMC)	Savitzky-Golay filter	hierarchical bidirectional deep RNN	–	–	Connectionist Temporal Classification (CTC)	94.5%
D. Rathi (2018) [31]	Camera	MobileNet , inception V3	Deepwise Separable CNN	Cross Entropy Loss	–	Softmax Function	95.03% & 93.32
Kang and et all (2015) [32]	Creative Senz3D camera	CaffeNet/subtract the mean image	CNN Depth map	–	–	Relu(Each layer)	85.49%
Masood and et all (2017) [33]	ILSVRC Dataset	VGG16 model /data augmentation	Deep CNN	–	Stochastic Gradient Descent	Softmax Function	96%
Arif-Ul-Islam and et all (2018) [34]	Microsoft Kinect sensor	Orientation based Hashcode & Gabor filter	ANN	–	ADAM	RELU	95.8
Rathi and et al, l 2020 [35]	Massey University Dataset (Barczak et al.)	Random Gaussian Noise & 2-level ResNet50	CNN	Categorical cross entropy	SGD	Softmax Function	99.03%
Cui et al. (2017) [38]	RWTH-PHOENIX-Weather 2014	VGG-S/ GoogLeNet (ILSVRC-2014)	CNN-BLSTM		SGD & ADAM	Softmax Function & CTC	38.7 (WER)
Junfu et al. (2018) [39]	RWTH-PHOENIX-Weather	3D-ResNet	CNN & Dilated Convolutions		SGD & ADAM	Softmax Function & CTC	37.3 (WER)
GUO et al. (2018) [40]	Chinese sign language(CSL)	C3D model (Tran. D et al.)	(HLSTM) & 3D CNN	log-likelihood	SGD	–	92.4
	RWTH-PHOENIXWeather 2014		CNNBLSTM & HMM	–	–	Softmax Function	

(continued on next page)

Table 2 (*continued*)

Author/year	Data set/image acquisition	Preprocessing/Pre-trained	NN Model	Loss Function	Optimizer	Classifier	Accuracy
Koller et al. *(2017)* [41]		GoogLeNet (ImageNet data set)					26.8 (WER)
Tao W et al. (2018) [43]	Microsoft Kinect device	Band-pass filter and circular region	CNN	cross-entropy	–	softmax function	93%
Garcia et al. (2016) [46]	Surrey University and Massey University ASL datasets	GoogLeNet (ILSVRC2012)	CNN	Xavier initialization [56]	–	softmax-based loss function:	97
Liang et al. (2018) [50]	Microsoft Kinect sensors	median filtering, zero mean	3D-CNN	negative log-likelihood	SGD	Softmax classifier	83.66
Bheda et al. (2017)55	Standard camera	image's background-subtraction	CNN	Categorical Cross Entropy Loss	SGD	Softmax classifier	83.5
Molchanov et al. (2015) [21]	VIVA challenge's Dataset	nearest neighbor interpolation (NNI)	3D-CNN (HRN & LRN)	negative log-likelihood	SGD	Softmax classifier	77.5
Anantha rao et al. (2018) [23]	Selfie Camera	high-performance computing (HPC)	CNN	–	Sgd	Softmax classifier & RELU	92.88
Yang et al. (2017) [36]	Standard camera	Haar feature classifier,	CNN	log-likelihood function	ADAGRAD	softmax function	99.688
Koller et al. (2018) [37]	(a)RWTH-2012, (b)RWTH-2014 (c)SIGNUM	Pixel-wise mean to subtract training image & GoogLeNet	CNN-HMM	–	–	Softmax Function	(a)30.0 (WER) (b)31.6 (WER) (c)7.4(W

Visual Recognition Challenge (ILSVRC)-2010 [33], ChaLearn Looking at People 2014 (CLAP14) [84], SIGNUM [37].

8. Conclusion

In this paper, we shared a quantitative study of different methods used in sign language recognition, covering 80 publications from 2010 until 2021. An analysis based on appearance-based SLR and vision-based SLR (deep learning) Each category was examined with roughly 40 papers. The following findings were observed in a study of papers:

A sign language recognition system has been developed from classifying only static signs and alphabets to a system that can effectively apprehend dynamic actions that come in continuous sequences of images.

Most papers published with vision-based approaches provide better results than appearance-based approaches. Researchers are currently paying more attention to making a large vocabulary for sign language recognition systems. Dataset availability and improvements in computing speed provide access to more training for given samples.

Many researchers are developing their SLR by using self-made small datasets. For some countries and languages, large datasets are still not available. The variant of sign language in most countries is based totally on their grammar and the way they provide each phrase, such as by presenting the language by using words or sentences.

The classification technique for identifying sign language also varies among researchers. Using their ideas and limitations for the Sign Language Recognition System, the comparison of one method to another method is still subjective. Deep learning-based approaches like CNN, RNN, LSTM, and Bi-Directional LSTM Models provide good recognition accuracy in the sequence of images and video streams.

CRediT authorship contribution statement

S. Subburaj: Conceptualization, Methodology, Formal analysis, Writing – original draft. **S. Murugavalli:** Conceptualization, Investigation, Supervision, Writing – review & editing.

Declaration of competing interest

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

References

[1] P.C. Badhe, V. Kulkarni, Indian sign language translator using gesture recognition algorithm, in: 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar, 2015, pp. 195–200.

[2] P.A. Nanivadekar, V. Kulkarni, Indian sign language recognition: database creation, hand tracking and segmentation, in: 2014 International Conference on Circuits, Systems, Communication and Information Technology Applications, CSCITA, Mumbai, 2014, pp. 358–363.

[3] Nikhil Kasukurthi, Brij Rokad, Shiv Bidani, Aju Dennisan American Sign Language Alphabet Recognition Using Deep Learning, 2014.

[4] A. Nandy, J.S. Prasad, S. Mondal, P. Chakraborty, G.C. Nandi, Recognition of isolated indian sign language gesture in real time, Inf. Process. Manag. (2010) 102–107.

[5] P.V.V. Kishore, M.V.D. Prasad, D.A. Kumar, A.S.C.S. Sastry, Optical flow hand tracking and active contour hand shape features for continuous sign language recognition with artificial neural networks, in: 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, 2016, pp. 346–351.

[6] G.A. Rao, P.V.V. Kishore, Selfie sign language recognition with multiple features on adaboost multilabel multiclass classifier, J. Eng. Sci. Technol. 13 (8) (2018) 2352–2368.

[7] P.V.V. Kishore, P.R. Kumar, A video based Indian Sign Language Recognition System (INSLR) using wavelet transform and fuzzy logic, Int. J. Eng. Technol. 4 (5) (2012) 537.

[8] B. Shi, et al., American sign language fingerspelling recognition in the wild, in: 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 2018, pp. 145–152.

[9] S.S. Shivashankara, S. Srinath, American sign language recognition system: an optimal approach, Int. J. Image Graph. Signal Process. (2018).

[10] Kshitij Bantupalli, Ying Xie, American sign language recognition using machine learning and computer vision, Master of Science in Computer Science Theses 21 (2019).

[11] M. Krishnaveni, V. Radha, Classifier fusion based on Bayes aggregation method for Indian sign language datasets, Procedia Eng. 30 (2012) 1110–1118.

[12] Yang Su, Qing Zhu, Continuous Chinese sign language recognition with CNN-LSTM, in: Proc. SPIE 10420, Ninth International Conference on Digital Image Processing (ICDIP 2017), 21 July 2017, p. 104200F, https://doi.org/10.1117/12.2281671.

[13] Q. Xiao, Y. Zhao, W. Huan, Multi-sensor data fusion for sign language recognition based on dynamic Bayesian network and convolutional neural network, Multimed. Tool. Appl. 78 (2019) 15335–15352, https://doi.org/10.1007/s11042-018-6939-8.

[14] S.G.M. Almeida, F.G. Guimarães, J.A. Ramírez, Feature extraction in brazilian sign language recognition based on phonological structure and using RGB-d sensors, Expert Syst. Appl. 41 (16) (2014) 7259–7271, https://doi.org/10.1016/j.eswa.2014.

[15] A. Wadhawan, P. Kumar, Deep Learning-Based Sign Language Recognition System for Static Signs, Neural Comput & Applic, 2020, https://doi.org/10.1007/s00521-019-04691-y.

[16] N.C. Camgoz, S. Hadfield, O. Koller, H. Ney, R. Bowden, Neural sign language translation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7784–7793.

[17] H. Lilha, D. Shivmurthy, Analysis of pixel level features in recognition of real life dual-handed sign language data set, in: Recent Trends in Information Systems (ReTIS), 2011 International Conference on, IEEE, 2011, December, pp. 246–251.

[18] N.C. Camgoz, S. Hadfield, O. Koller, R. Bowden, SubUNets: end-to-end hand shape and continuous sign language recognition, in: 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 3075–3084.

[19] A. Kika, A. Koni, Hand gesture recognition using convolutional neural network and histogram of oriented gradients features, in: CEUR Workshop Proceedings, vol. 2280, CEUR-WS, 2018, pp. 75–79.

[20] R. Akmeliawati, M.P. Ooi, Y.C. Kuang, Real-time Malaysian sign language translation using colour segmentation and neural network, in: 2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007, Warsaw, 2007, pp. 1–6.

[21] P. Molchanov, S. Gupta, K. Kim, J. Kautz, Hand gesture recognition with 3D convolutional neural networks, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, Boston, MA, 2015, pp. 1–7, https://doi.org/10.1109/CVPRW.2015.7301342.

[22] J. Nagi, et al., Max-pooling convolutional neural networks for vision-based hand gesture recognition, in: 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, 2011, pp. 342–347, https://doi.org/10.1109/ICSIPA.2011.6144164.

[23] G.A. Rao, K. Syamala, P.V.V. Kishore, A.S.C.S. Sastry, Deep convolutional neural networks for sign language recognition, in: 2018 Conference on Signal Processing and Communication Engineering Systems (SPACES), Vijayawada, 2018, pp. 194–197, https://doi.org/10.1109/SPACES.2018.8316344.

[24] Biyi Fang, Jillian Co, Mi Zhang, DeepASL: enabling ubiquitous and non-intrusive word and sentence-level sign language translation, in: Proceedings of the 15th ACM Conference on Embedded Networked Sensor Systems (SenSys '17), 2017.

[25] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, Woosub Jung, SignFi: sign language recognition using WiFi, Proc. ACM Interact. Mob. Wear. Ubiq. Technol. 2 (1) (2018) 21. Article 23 (Mar. 2018).

[26] P.V.V. Kishore, K.B.N.S.K. Chaitanya, G.S.S. Shravani, Teja Maddala, Kiran Eepuri, D. Anil Kumar, DSLR-net a Depth Based Sign Language Recognition Using Two Stream Convents, vol. 8, 2019, pp. 765–773.

[27] Srujana Gattupalli, Amir Ghaderi, Vassilis Athitsos, Evaluation of deep learning based pose estimation for sign language recognition, in: Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '16), Association for Computing Machinery, New York, NY, USA, 2016, https://doi.org/10.1145/2910674.2910716. Article 12, 1–7.

[28] D. Avola, M. Bernardi, L. Cinque, G.L. Foresti, C. Massaroni, Exploiting recurrent neural networks and Leap motion controller for the recognition of sign language and semaphoric hand gestures, IEEE Trans. Multimed. 21 (1) (Jan. 2019) 234–245.

[29] Jie Huang, Wengang Zhou, Houqiang Li, Weiping Li, Sign Language Recognition using 3D convolutional neural networks, in: 2015 IEEE International Conference on Multimedia and Expo, ICME, Turin, 2015, pp. 1–6.

[30] Bowen Shi, Aurora Martinez Del Rio, Jonathan Keane, Diane Brentari, Greg Shakhnarovich, Karen Livescu, Fingerspelling Recognition in the Wild with Iterative Visual Attention, 2019.

[31] D. Rathi, Optimization of Transfer Learning for Sign Language Recognition Targeting Mobile Platform, 2018 arXiv preprint arXiv:1805.06618.

[32] B. Kang, S. Tripathi, T. Nguyen, "Real-time sign language fingerspelling recognition using convolutional neural networks from depth map", Pattern Recognition 2015 3rd IAPR Asian Conference on, Nov. 2015.

[33] S. Masood, H.C. Thuwal, A. Srivastava, S. Satapathy, V. Bhateja, S. Das, American sign language character recognition using convolution neural network, in: Smart Computing and Informatics. Smart Innovation Systems and Technologies, vol. 78, Springer, Singapore, 2018.

[34] Arif-Ul-Islam, S. Akhter, Orientation hashcode and articial neural network based combined approach to recognize sign language, in: 2018 21st International Conference of Computer and Information Technology, ICCIT, Dhaka, Bangladesh, 2018, pp. 1–5.

[35] Pulkit Rathi, Kuwar Gupta, Raj, Soumya Agarwal, Anupam Shukla, sign language recognition using ResNet50 deep neural network architecture. https://doi.org/10.2139/ssrn.3545064, February 27, 2020.

[36] S. Yang, Q. Zhu, Video-based Chinese sign language recognition using convolutional neural network, in: 2017 IEEE 9th International Conference on Communication Software and Networks, ICCSN, Guangzhou, 2017, pp. 929–934, https://doi.org/10.1109/ICCSN.2017.8230247.

[37] O. Koller, S. Zargaran, H. Ney, et al., Deep sign: enabling Robust statistical continuous sign language recognition via hybrid CNN-HMMs, Int. J. Comput. Vis. 126 (2018) 1311–1325, https://doi.org/10.1007/s11263-018-1121-3.

[38] R. Cui, H. Liu, C. Zhang, Recurrent convolutional neural networks for continuous sign language recognition by staged optimization, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, 2017, pp. 1610–1618.

[39] Junfu Pu, Wengang Zhou, Houqiang Li, Dilated convolutional network with iterative optimization for coutinuous sign language recognition, in: International Joint Conference on Artificial Intelligence, IJCAI, 2018, pp. 885–891.

[40] D. Guo, W. Zhou, H. Li, M. Wang, Hierarchical LSTM for sign language translation, in: AAAI Conference on Artificial Intelligence, North America, apr. 2018.

[41] O. Koller, S. Zargaran, H. Ney, Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, 2017, pp. 3416–3424.

[42] Eriglen Gani, Alda Kika, Albanian sign language (AlbSL) number recognition from both hand's gestures acquired by Kinect sensors, Int. J. Adv. Comput. Sci. Appl. 7 (2016) 7, 2016.

[43] W. Tao, M.C. Leu, Z. Yin, American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion, Eng. Appl. Artif. Intell. 76 (2018) 202–213.

[44] A.A. Ahmed, S. Aly, Appearance-based Arabic sign language recognition using hidden Markov models, in: 2014 International Conference on Engineering and Technology, ICET, Cairo, 2014, pp. 1–6.

[45] C. Savur, F. Sahin, American Sign Language Recognition system by using surface EMG signal, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC, Budapest, 2016, 002872-002877.

[46] Brandon Garcia, Sigberto Viesca, Real-time American sign language recognition with convolutional neural networks, in: Convolutional Neural Networks for Visual Recognition at Stanford University, 2016.

[47] M. Boulares, M. Jemni, 3D motion trajectory analysis approach to improve sign language 3d-based content recognition, Procedia Comput. Sci. 13 (2012) 133–143.

[48] T. Raghuveera, R. Deepthi, R. Mangalashri, et al., A depth-based Indian sign language recognition using Microsoft Kinect, Sādhanā 45 (2020) 34, https://doi.org/10.1007/s12046-019-1250-6.

[49] K.M. Lim, A.W.C. Tan, S. Tan, A feature covariance matrix with serial particle filter for isolated sign language recognition, Expert Syst. Appl. 54 (2016) 208–218, https://doi.org/10.1016/j.eswa.2016.01.047.

[50] Zhi-jie Liang, Sheng-bin Liao, Bing-zhang Hu, 3D convolutional neural networks for dynamic sign language recognition, Comput. J. 61 (11) (November 2018) 1724–1736, https://doi.org/10.1093/comjnl/bxy049.

[51] Rúbia Reis Guerra, Rezende, Tamires Martins Guimarães, Frederico Gadelha, Sílvia Grasiella Moreira Almeida, Facial expression analysis in Brazilian sign language for sign recognition, in: NATIONAL MEETING OF ARTIFICIAL AND COMPUTATIONAL INTELLIGENCE, ENIAC, 2018.

[52] P.K. Athira, C.J. Sruthi, A. Lijiya, A signer independent sign language recognition with co-articulation elimination from live videos: an indian scenario, J. King Saud Univ. Comput. Inf. Sci. (2019), https://doi.org/10.1016/j.jksuci.2019.05.002.

[53] M. Mohandes, M. Deriche, U. Johar, S. Ilyas, A signer-independent Arabic sign language recognition system using face detection, geometric features, and a hidden Markov model, Comput. Electr. Eng. 38 (2) (2012) 422–433, https://doi.org/10.1016/j.compeleceng.2011.10.013.

[54] B.M. Chethana Kumara, H.S. Nagendraswamy, R Lekha Chinmayi, Spatial relationship based features for Indian sign language recognition, International Journal of Computing, communications & Instrumentation Engineering 3 (2) (2016), 2349- 1469.

[55] V. Bheda, D. Radpour, Using Deep Convolutional Networks for Gesture Recognition in American Sign Language, 2017 arXiv preprint arXiv:1710.06836.

[56] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: International Conference on Artificial Intelligence and Statistics, 2010.

[57] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, H. Ney, Extensions of the sign language recognition and translation Corpus RWTH-PHOENIX-weather, in: International Conference on Language Resources and Evaluation, LREC, 2014.

[58] Yannis M. Assael, Brendan Shillingford, Shimon Whiteson, Nando de Freitas, LipNet: Sentence-Level Lipreading, 2016 arXiv preprint arXiv:1611.01599 (2016).

[59] Christian Szegedy, Vanhoucke Vincent, Sergey Ioffe, Jonathon Shlens, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, 2015, pp. 2818–2826.

[60] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Weyand Tobias, Marco Andreetto, Hartwig Adam, MobileNets: efficient convolutional neural networks for mobile vision applications. https://arxiv.org/abs/1704.04861, 2017.

[61] http://www.massey.ac.nz/~albarcza/gesture_dataset2012.html.

[62] S.J. Pan, Q. Yang, et al., A Survey on Transfer Learning, IEEE Transactions on knowledge and data engineering, 2010.

[63] sign language MNIST, Kaggle. https://www.kaggle.com/datamunge/sign-language-mnist/, 2017.

[64] Y. Jia, Caffe: an open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org, 2013.

[65] Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014, p. 1556, arXiv preprint arXiv:1409.

[66] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Reed Scott, Dragomir Anguelov, Dumitru Erhan, Vanhoucke Vincent, Andrew Rabinovich, Going deeper with convolutions, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 1–9. Boston, Ma, USA.

[67] http://vlm1.uta.edu/~srujana/ASLID/ASL_Image_Dataset.html.

[68] Alex Graves, Fernández Santiago, Faustino Gomez, Jürgen Schmidhuber, Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 369–376.

[69] Yong Du, Wei Wang, Liang Wang, Hierarchical recurrent neural network for skeleton based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1110–1118.

[70] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: ICCV, 2015, pp. 4489–4497 [40].

[71] Alexander Toshev, Christian Szegedy, Deeppose: human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[72] Sylvie C.W. Ong, Surendra Ranganath, Automatic sign language analysis: a survey and the future beyond lexical meaning, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 6, https://doi.org/10.1109/TPAMI.2005.112 (June 2005), 873–891.

[73] M. Eslami, M. Karami, S. Tabarestani, F. Torkamani-Azar, S. Eslami, C. Meinel, SignCol: open-source software for collecting sign language gestures, in: 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2018, pp. 365–369.

[74] Dong Cao, M.C. Leu, Z. Yin, American sign language alphabet recognition using Microsoft Kinect, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, Boston, MA, 2015, pp. 44–52.

[75] Becky Sue Parton, sign language recognition and translation: a multidisciplined approach from the field of artificial intelligence, J. Deaf Stud. Deaf Educ. 11 (1) (2006) 94–101, https://doi.org/10.1093/deafed/enj003. Winter.

[76] D. Soydaner, A comparison of optimization algorithms for deep learning, Int. J. Pattern Recogn. Artif. Intell. 34 (13) (2020), 2052013.

[77] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, ICLR, 2015.

[78] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[79] T. Liu, W. Zhou, H. Li, Sign Language Recognition with long short-term memory, in: 2016 IEEE International Conference on Image Processing, ICIP, Phoenix, AZ, 2016, pp. 2871–2875.

[80] Y. Liao, P. Xiong, W. Min, W. Min, J. Lu, Dynamic Sign Language Recognition based on video sequence with BLSTM-3D residual networks, IEEE Access 7 (2019) 38044–38054.

[81] S. Yang, Q. Zhu, Video-based Chinese Sign Language Recognition using convolutional neural network, in: IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, 2017, pp. 929–934.

[82] https://www.site.uottawa.ca/research/viva/projects/hand_detection/index.html.

[83] https://image-net.org/challenges/LSVRC/2010/.

[84] https://gesture.chalearn.org/2014-looking-at-people-challenge.

[85] https://www-i6.informatik.rwth-aachen.de/~koller/1miohands-data/.

[86] https://www.phonetik.uni-muenchen.de/forschung/Bas/SIGNUM/.

[87] https://asluniversity.com/.

[88] https://www-i6.informatik.rwth-aachen.de/aslr/database-rwth-boston-50.php.

[89] https://www.massey.ac.nz/~albarcza/gesture_dataset2012.html.