Computers, Materials & Continua
DOI:10.32604/cmc.2020.013905
Article

Tech Science Press

Intelligent Dynamic Gesture Recognition Using CNN Empowered by Edit Distance

Shazia Saqib[1], Allah Ditta[2], Muhammad Adnan Khan[1,*], Syed Asad Raza Kazmi[3] and Hani Alquhayz[4]

[1]Department of Computer Science, Lahore Garrison University, Lahore, 54000, Pakistan
[2]Department of Information Sciences, Division of Science & Technology, University of Education, Lahore, 54000, Pakistan
[3]GC University, Lahore, 54000, Pakistan
[4]Department of Computer Science and Information, College of Science in Zulfi, Majmaah University, Al-Majmaah, 11952, Saudi Arabia
[*]Corresponding Author: Muhammad Adnan Khan. Email: madnankhan@lgu.edu.pk
Received: 26 August 2020; Accepted: 28 September 2020

Abstract: Human activity detection and recognition is a challenging task. Video surveillance can benefit greatly by advances in Internet of Things (IoT) and cloud computing. Artificial intelligence IoT (AIoT) based devices form the basis of a smart city. The research presents Intelligent dynamic gesture recognition (IDGR) using a Convolutional neural network (CNN) empowered by edit distance for video recognition. The proposed system has been evaluated using AIoT enabled devices for static and dynamic gestures of Pakistani sign language (PSL). However, the proposed methodology can work efficiently for any type of video. The proposed research concludes that deep learning and convolutional neural networks give a most appropriate solution retaining discriminative and dynamic information of the input action. The research proposes recognition of dynamic gestures using image recognition of the keyframes based on CNN extracted from the human activity. Edit distance is used to find out the label of the word to which those sets of frames belong to. The simulation results have shown that at 400 videos per human action, 100 epochs, 234 × 234 image size, the accuracy of the system is 90.79%, which is a reasonable accuracy for a relatively small dataset as compared to the previously published techniques.

Keywords: Sign languages; keyframe; edit distance; misrate; accuracy

1 Introduction

Video content (a sequence of 2D frames) is globally growing exponentially every year. As a result, lots of effort has been made in the image and video recognition domain. Video classification and video captioning are two major active research areas at the moment. Video classification recognizes these videos using their content while the video captioning gives a short description of these videos using their content. Video classification is done in the spatial domain as well as in the temporal domain either separately or collectively. Convolutional neural networks (CNN) has given promising performance for analyzing image

content, image recognition, detection, and retrieval. These networks can process millions of parameters and handle huge labelled datasets for learning. This has led to the testing of CNN in large scale video classification, in static images as well as in complex temporal evolution of the videos.

Processing raw video sequences are not efficient as they have very high dimensionality depending on image dimensions and video duration. A video is a sequence of images. Most images in a video do not contain any new information. These images keep repeating, and usually after 10–15 frames approximately, a new chunk of data from the video appears and is vital for action recognition [1]. This leads to the use of keyframes for action recognition. The keyframe represents valuable information in that temporal segment. The edit distance identifies the class of the video using the summarized keyframes. To recognize videos, several supervised and unsupervised techniques have been used that are based on bio-inspired sign descriptors, border(boundary) following, chain codes, polygonal approximation, Fourier descriptors, polygonal approximation, Fourier descriptors, statistical descriptors, regional descriptors, and deep learning [2]. However, deep learning-based techniques have given better results than all other techniques.

CNN can perform well if the system works with reliable datasets and GPUs. But still, many issues remain to make the system robust and practical. These are some of the problems:

1.1 Huge Datasets Required

All recognition systems depend on the extensive collection of videos. In many situations, a large video training set may not be available, So this puts some limitations on the use of CNN for recognition systems. We need to work with networks that can give good results with reasonably sized training data.

1.2 Invariance

The recognition systems must be invariant to translation rotation and scaling. While dealing with video invariances in 3D is needed.

1.3 Handling Degradations in Training Data

The networks should be robust to low resolution, blurring, pose variations, illumination, and occlusion [3].

1.4 Structure of the Network

The decisions like the number of layers, fully connected layers, dropouts, max-pooling operations can affect the efficiency of the CNN [3].

1.5 Training and Validation Set Generation

To determine the performance of the network, we divide our video data set to training validation and testing.

1.6 Early Fusion

The early fusion methods combine input features from various modalities. The fusion is done immediately on the lowest possible level, which is a pixel level. The network learns the correlation and interactions of each modality. Early fusion performs multimodal learning. It usually requires the features from different modalities to align with their semantics. It uses a single model to predict, which shows that the model is well suited for all the modalities. The early and direct connectivity to pixel data allows the network to detect local motion speed and direction [3] precisely.

1.7 Exploding Vanishing Gradients

The problem requires to use low learning rates with gradient descent. For a slow computer, this process will take a long time for each step. A faster GPU can overcome this delay. Another way to handle this problem is to add more hidden layers which help the network to learn more complex arbitrary functions, and in predicting future outcomes.

The paper layout is as explained: Section 2 shows previous work done in the past in this domain, Section 3 elaborates on the experimental work based on the algorithm written in Section 3. Section 4 discusses the outcomes of the experiment, Section 6 compares the proposed system with the existing techniques and Section 7 gives a conclusion and suggests future work.

2 Related Work

Kanehira et al. [1] proposed Fisher's discriminant criteria for an inner summary, inner group, and between-group variances defined on the feature representation of summary. SE De Avila et al. [2] proposed Video summarization (VSUMM) for producing static video summaries using the k-means clustering algorithm. Sebastian et al. [3] have used the mean, variance, skew, and kurtosis histogram of every block and compared it with the corresponding blocks of the next frame. Kamoji et al. [4] have analyzed the motion, block matching techniques based on diamond search, and three-step search. Gong et al. [5] has used the Sequential determinantal point process (SEQDPP) for keyframe selection based on a random permutation of video frames.

Cahuina et al. [6] have proposed a technique of using local descriptors for semantic video summarization. They tested the method on 100 videos. Shi et al. [7] have proposed a keyframe extraction method for video copyright protection. Mahmoud et al. [8] have suggested the use of VGRAPH that used color as well as texture features. Guan et al. [9] have suggested a key point-based framework to select keyframes using local features. Asade et al. [10] suggested an algorithm to extract static video summaries using fuzzy c-means clustering. Kim et al. [11] have proposed a technique that generates panoramic images from web-based geographic information systems using data fusion, crowdsourcing, and recent advances in media processing. Danelljan et al. [12] used Discriminative correlation filters (DCF) for visual object tracking. Wang et al. [13] proposed dense trajectories to recognize an action in videos. Surf descriptors and dense optical flow were used to compare feature points for estimating homographs. This significantly improves motion-based descriptors, such as Histograms of optical flow (HOF) and Motion boundary histograms (MBH). Experimental results on four challenging action datasets (i.e., Hollywood2, HMDB51, Olympic Sports, and UCF50) give better results than other techniques. Bansal et al. [14] have used the hidden Markov model as an indispensable tool for the recognition of dynamic gestures in real-time. Bhuyan [15] has proposed gesture spotting to eliminate the effects of changes in a "Motion chain code (MCC)". Kalman filter determines the track of each person [16]. Mei et al. [17] proposed a constrained Minimum sparse reconstruction (MSR) model-based Video summarization (VS). Muhammad et al. [18] have used an effective shot segmentation method based on deep features. They have used entropy along with memorability testing their algorithm on two video datasets. Burec et al. [19] have used models inspired by human models using estimation of joint trajectories, and spatiotemporal local descriptors.

Panda et al. [20] has used graph clustering based on random walks using a factor-based ranking. Voulodimoswe et al. [21] have used k means ++ clustering as well as temporal summaries on two dance motion datasets and got promising results. Zhang et al. [22] have used semantic side information of video to generate subshot-based summarization at significantly less computational costs testing their work on several benchmarks and got promising results. Chellappa et al. [23] has used Deep convolutional neural networks (DCNNs) for face and other objects recognition giving promising results.

Singha et al. [24] has developed a classifier fusion based dynamic freehand gesture recognition system using a two-level speed normalization procedure based on Dynamic time warping (DTW) and Euclidean distance. Pigou et al. [25] proposed a technique that uses a simple pooling strategy using the temporal aspect of the video. Varol et al. [26] have used Long term temporal convolutions (LTC). They proved that LTC-CNN models give an increased accuracy of action recognition. Jiang et al. [27] have proposed a framework using the feature relationships and the class relationships by imposing regularization thus offering Regularized deep neural networks (rDNN) for modelling video semantics getting reasonable results on Hollywood2 and Columbia video benchmarks. Donahue et al. [28] have used recurrent convolutional architectures for image captioning, activity recognition, and video description. Simonyan et al. [29] have proposed a spatiotemporal ConvNet architecture that uses multi-frame dense optical flow on limited training data. They tested their technique on CF-101 and HMDB-51. Tran et al. [30] proposed architecture for spatiotemporal feature learning using 3 3 3 convolution kernels. It gives the best architectures for 3D ConvNets. The learned Convolutional3D (C3D) features along with a linear classifier giving 52.8% accuracy on UCF101 dataset. Thakre et al. [31] have made use of video partitioning and keyframe extraction for video analysis and content-based video retrieval. Sheena et al. [32] have proposed a method that uses the difference of histograms in consecutive frames calculates the mean and standard deviation of the difference between frames. Then using these values threshold is calculated. The experiments are conducted on the KTH action database. Ng et al. [33] have used Long Short Term Memory (LSTM) cells based Recurrent Neural Network. Lillicrap et al. [34] have worked on Deep Q-learning, a technique, based on the "deterministic policy gradient". Redmon et al. [35] have a deep learning neural network called YOLO. Ren et al. [36] have used the region proposal algorithms to identify object locations called Region proposal network (RPN). Nam et al. [37] have worked on Convolutional neural networks (CNNs). Bertinetto et al. [38] has used Stochastic gradient descent to adjust the weights of the network, compromising the speed of the system. Feichtenhofer et al. [39] have suggested ways of fusing ConvNet towers in the spatial and the temporal domain.

Zhu et al. [40] has proposed a full "visual tracking procedure" in videos using "Reinforcement Learning Algorithms". Song et al. [41] have presented Title-based video summarization (TVSUM). Zaal et al. [42] have used algorithm Iterative self-organizing data analysis technique (ISODATA) to cluster frames into classes automatically. Saqib et al. [43] has proposed a method for video summarization based on entropy and mean of frames which use human activity recognition in place of the full video. Ejaz et al. [44] have combined the features of Red Green Blue (RGB) color channels, histograms, and moments to find the keyframes. The technique is adaptive as it combines current and old iterations. Jaouedi et al. [45] have used hybrid deep learning based on Gated recurrent neural networks (GRNN) model for human action recognition tested on UCF Sports, UCF101, and KTH datasets. The research analyses videos and extract related features using GMM and KF methods. The visual characteristic of each frame from the input video is used along with recurrent neural networks model based on the gated recurrent unit. The research analyses and extracts all features in all frames of video. The research applies to a wide range of applications.

3 Proposed Solution

The proposed technique is based on the Convolution neural network(CNN). Fig. 1 shows a transition from pixel to actual object recognition using 3 hidden layers. The first hidden layer finds the edges, 2nd layer finds the contours, 3rd layer detects the parts of the body. The recognition process evolves from edge detection to contours detection at the next layer leading to parts detection at the next hidden layer. This in turn, leads to object detection at the last output layer.

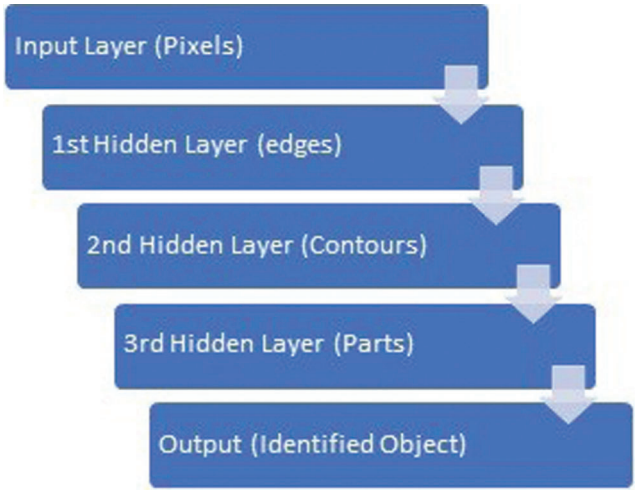

Figure 1: A convultional network

The layers in CNN use the features learned by the preceding layers to recognize the larger patterns. The classification layer combines them to group the images. Its output is equal to the number of classes in the target data. The sign language used in this research is Pakistan sign language (PSL).

The classification is done using the softmax function. The output by the softmax activation function helps in dividing each input to its corresponding classes. Accuracy is the measure of the number of true labels in the test data. Using the training data, CNN understands the object's specific features and associates them with the corresponding category. Layers get data from the previous layer, process it, and pass it on. The network learns features of images on its own. The entire cycle starts with capturing input video, dividing it into frames of order 1280 × 720, and selecting the keyframes. The input is human action in the form of a video. The video is converted to sequential frames $f_1\ f_2\ f_3\ \ldots\ f_n$. The system selects keyframes using the Median of Entropy of Mean Frames Method [43]. These keyframes are recognized using CNN.

The class to which these keyframes belong to forms an output string. The string is fed into the edit distance algorithm to find out the closest matching word. The layers in CNN use the features learned by the preceding layers to recognize the larger patterns. The classification layer combines them to group the images. Its output is equal to the number of classes in the target data. The sign language used in this research is PSL. The Input Layer is where we specify the image size for the images extracted as keyframe from the input video, which, in this case, is 234, and channel size is 1 as the images are in grayscale colors. The convolutional layer specifies filter size, which is the height and width of the filters during the training phase moved along the images extracted as a keyframe from input videos. We can use different sizes for the height and the width of the filter. Another feature is the number of filters, which specifies the number of neurons connecting to the same output area. This convolution layer determines the number of feature maps. The strides are taken as 1 for the convolution layer. The learning rate for this layer is kept relatively low.

Fig. 2 shows a complete architecture of the process of human action recognition. The ReLU Layer introduces nonlinearity in the neural network. The network uses max pooling for downsampling operation to reduce the number of parameters. This layer returns the maximum of a region of 2 × 2. The fully connected layer follows all the convolutional layers. All neurons in a fully connected layer are connected to the neurons in the previous layer. The last fully-connected layer combines them to classify the images. The fully connected layer usually uses the softmax activation function for classification. The last layer is

the classification layer which uses the probabilities returned by the softmax activation function for each input to determine the output classes. The results also show the mini-batch loss and the mini-batch accuracy for the first iteration, last iteration, and every 50 iterations in between. The mini-batch loss is also called the cross-entropy loss. The mini-batch accuracy is the percentage of images in the current mini-batch that the network being trained correctly classifies. It also returns the cumulative time it takes for training. Testing accuracy is a measure of the number of true labels in the test data. Using the training data, CNN understands the object's specific features and associates them with the corresponding category. Layers get data from the previous layer, process it, and pass it on. The network learns features of images on its own, and we have no role in that.

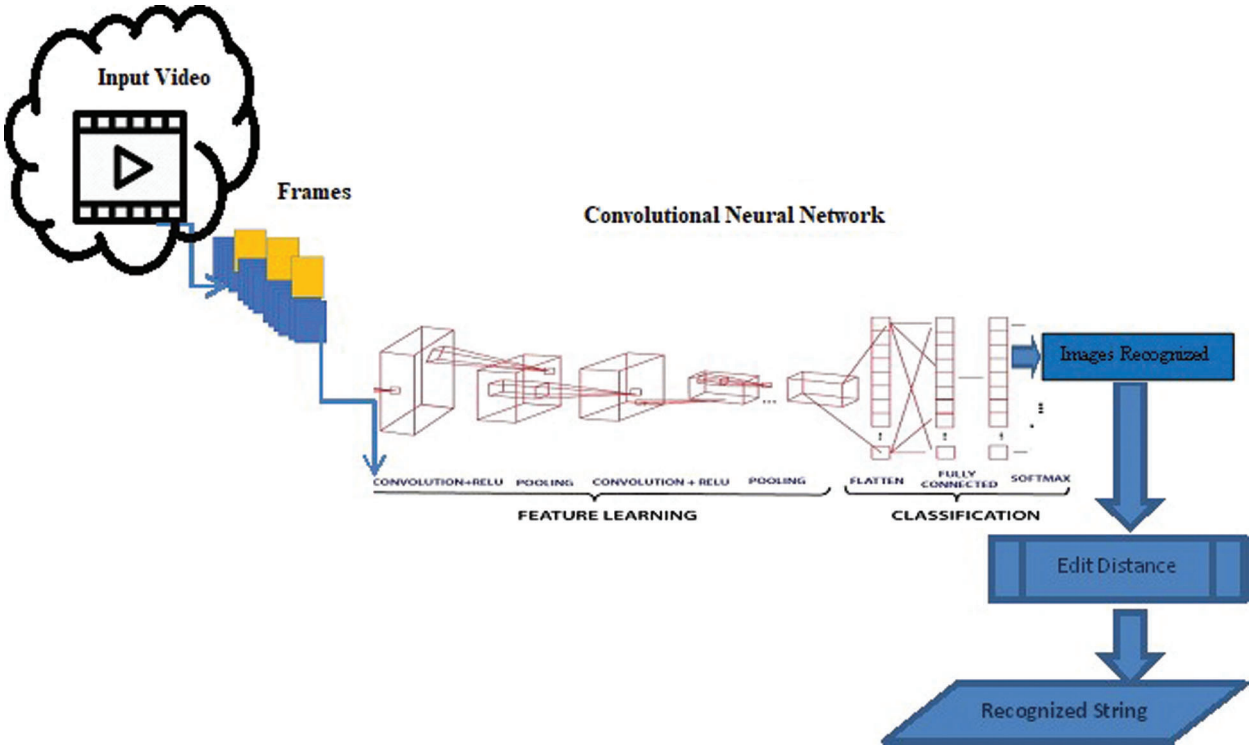

Figure 2: System architecture for human action recognition

The constraint $a\ (\alpha_1\ (t) + \alpha_2\ (t)) = a\ (\alpha_1\ (t)) + a(\alpha_2\ (t))$ ensures linearity and the constraint $\alpha\ (t) = \alpha\ (t,\ t_0)$ ensures time invariance. Here an input function $\alpha\ (t)$ is combined with a function $\eta(t)$ to generate output that signifies convolution as a mathematical operation performed on Linear Time-Invariant (LTI) systems. The overlapping $\alpha(t)$ and the reverse of $\eta(t)$ function. This function $\eta(t)$ is the filter or kernel transformation. We define the output β as follows:

$$\beta(I,j) = (\eta \times \alpha)(I,j) \sum_{n} \sum_{m} \eta(m,n)\alpha(i-m,\ j-n)$$

The inputs are zero-padded at the edges, to help filters fit near the edges. The number of zeros involved in zero-padding units is another hyperparameter to improve efficiency.

It should also be ensured to match the number of channels in the filters as well as the number of channels in its input. The convolution layer outputs go into a nonlinear layer/stage, which is just like the activation function. The detector layer normally uses the sigmoid function or hyperbolic tangent tanh ReLU for

inducing nonlinearity in the model. A CNN block consists of one convolutional layer, an activation function like ReLU, and a pooling layer combined to form a network layer. The output of these blocks is flattened and sent to a fully connected output layer.

3.1 Algorithmic Solution for Image Recognition

Create a datastore of images Store $X_{i\,j}$ in subfolders in datastore For each image I in a subfolder, convert all images from RGB to grayscale. Resize each image to size J × K.

The algorithm is as under:

Trainingpercentage ← 81

Testingandvalidationpercentage ← 19

ImageInputLayer ← 1

MaxPooling2dLayer ← 1

ClassificationLayer ← 1

Filtersize ← f

Number of filters ← n

Epochs ← defined n of epochs

learningrate ← .00001

TraintTheSystem()

Accuracy ← trainingimagesmatched ÷ totalimages

Misrate ← trainingimagesmismatched ÷ totalimages

- The input layer uses only grayscale pixel values and is of sizes as shown in Tab. 1.
- A filter of 5 with no padding and stride s = 1 is used. A maxpooling layer with pool size is used with a pooling stride = 2 for maxpooling . The experiment is repeated will filter of size 3 × 3, 5 × 5 and 7 × 7. As the images are in grayscale, channel size is one. The number of filters are varied from 20 to 30 in the experiment. A fully connected layer follows it.
- ReLu function is used to introduce nonlinearity in the model.
- A maxpooling layer is introduced with stride = 2.
- At the end of the network, a softmax layer and a classification layer is used to determine cross-entropy loss for the proposed solution.
- The learning rate is kept as low as 0.00001.

3.2 Video Dataset

Several video datasets are publicly available, however, for this research, the video dataset is formed by selecting 20 words from Pakistan sign language. 15 signers participated in preparing videos for 20 words. Approximately 400 videos for every word gesture are collected. The videos are preprocessed and passed through the video summarization process. The converted images will be stored in subfolders under that word gesture folder. The images are in grayscale and of size 234 × 234. This dataset has 8,000 video clips from 20 different categories prepared for 20 words by 15 different signers. The duration of these is 11.2 hours approximately. We use these 400 × 20 videos by 15 signers to train, validate, and test the network.

The video recognition process has two main components: Video summarization and image recognition. The process of video summarization consists of selecting keyframes, meaningful clip selection, and output generation. The technique proposed here uses the concept of mean and then median of entropy. The mean is a

very important measure in digital image processing. It is used in spatial filtering and is helpful in noise reduction. The mean of k frames is defined as:

Table 1: Result of applying different layers of CNN

Resolution	Dataset size	Epoch	Accuracy%	Miss rate%
72 × 72	200	15	45.91	54.09
72 × 72	200	50	55.69	44.31
72 × 72	400	50	87.06	12.94
72 × 72	400	100	89.22	10.78
90 × 90	200	100	85.4	14.6
90 × 90	300	100	87.4	12.6
90 × 90	400	100	90.4	9.6
100 × 100	200	100	86.35	13.65
100 × 100	300	100	87.6	12.4
100 × 100	400	100	89.15	10.85
120 × 120	200	15	70.33	29.67
120 × 120	200	50	86.34	13.66
120 × 120	300	100	88.38	11.62
120 × 120	400	50	89.59	13.41
120 × 120	400	100	90.53	9.47
234 × 234	300	100	88.69	11.31
234 × 234	400	15	80.69	19.31

$$\bar{f}(i,j) = \frac{\sum_{m=1}^{k}\sum_{i=1}^{N}\sum_{j=1}^{N} f_m(i,j)}{k}$$

Here $\bar{f}(i,j)$ shows mean of k images of size N × N.

$\sum_{m=1}^{k}\sum_{i=1}^{N}\sum_{j=1}^{N} f_m(i,j)$ is the sum of k frames.

$\sum_{i=1}^{N}\sum_{j=1}^{N} f_m(i,j)$ shows m^{th} frame.

A video summary is generated as under:

- The input video is the video that is to be used for video summarization, the video may be in any standard format.
- Frame extraction from videos as a finite number of still images called frames.
- The feature extraction process can be based on features like color, edge, or motion features. Some algorithms use other low-level features such as color histogram, frame correlation, and edge histogram.

The video is summarized to keyframes by using the technique Median of Entropy of Mean Frames Method [43]. The proposed algorithm can be used for any type of video however it has performed well for continuous gestures.

3.3 Edit Distance

The Levenshtein distance or edit distance was named after his inventor, Vladimir I. Levenshtein. The Levenshtein distance is the number of edits needed to convert a sequence A into another sequence B. Edit operation consists of substitutions, insertions, and deletions. The variation Damerau Levenshtein distance adds an extra root in dynamic programming. It computes a $d[I, j]$ which stores the edit distance between $a_1 \ldots a_i$ and edit operation which is a transposition. This operation interchanges two adjacent characters. To express the Damerau Levenshtein distance d between two strings a and b.

$$d_{a,b}(i,j) = min \begin{cases} 0 & if\ i = j = 0 \\ d_{a,b}(i-1,j) + 1 & if\ i > 0 \\ d_{a,b}(i,j-1) + 1 & if\ j > 0 \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_i)} & if\ i,j > 0 \\ d_{a,b}(i-2,j-2) + 1 & if\ i,j > 1\ and\ a[i] = b[j-1] and\ a[i-1] = b[j] \end{cases}$$

4 Video Recognization

The PSL is a very rich sign language. It consists of thousands of words. The video classification works precisely like the image classification, as explained in Section 3.1. The video is summarized, and individual frames are stored in that particular video category and frame number. At the same time, the labels of summarized frames are stored in repository DS. These frames are trained using the images in the dataset. The process is repeated for all the words in the dictionary, the summarized images are combined into folders containing similar images. The CNN model is trained by using frames in the dataset. In the test phase, the frames are used to predict the folder label. In test mode, every frame from the video summary is predicted for the category to which it belongs to. The output string consisting of the folder labels is compared with the strings in DS. The string with minimum edit distance is chosen as the output string. Algorithm of the recognized dynamic gesture given below.

Algorithm: Recognize the Dynamic Gesture

Input: The Video converted to $f_1, f_2, f_3 \ldots f_i$ *where* $1 \leq t \leq tkfr$

The datastore:DS[m] dictionary of m words containing at most size number of images

datastore: dgestures containing L folders

Output: wordrecognized

$v[i]$ = *label*(f_i) *using algorithm in Section* 3.1

$\forall f_i$ *where* $1 \leq i \leq tk\,f\,r$

comparevwithds[m] *using EditDistance* $\forall 1 \leq m \leq size$

wordrecognized = *DS*(*min*(*ED*(*DS, v*)))

5 Discussion

The dynamic sign recognition starts with image recognition. The labels of the recognized images help in identifying the dynamic gesture class, i.e., complete words using edit distance algorithm. The video recognition is analyzed as under:

5.1 Images

Tab. 1 shows the results of applying CNN to the dataset of hand gesture images. After going through numerous training and testing rounds of the system, the accuracy of image recognition is 91.03% as shown in Tab. 1 at 100 epochs, image size 234 × 234, and a dataset size equal to 400 which is a very reasonable rate of recognition. It can give much better results for the larger dataset and a higher number of epochs.

Fig. 3 shows a graph of image size, epochs, data set size, and accuracy. It shows that for fixed image size, the graph of the epoch, data set size and accuracy. Fig. 3 shows the proposed solution accuracy with respect to dataset size & Epochs. It is shown that if epochs are increased & the dataset size is kept constant, then the accuracy of the proposed solution is increased up to 91.01%. It observed that if dataset size is increased and so are the Epochs, system accuracy is also increased.

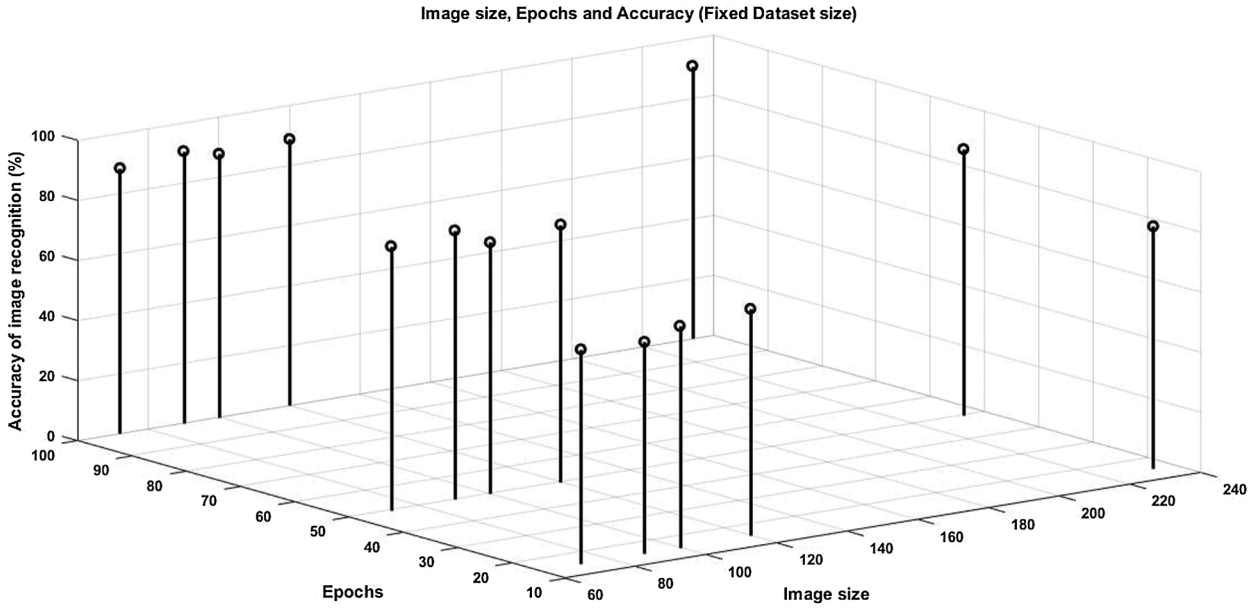

Figure 3: Accuracy of the proposed solution for fixed dataset size, variable epochs and variable image size

Fig. 4 shows the proposed solution accuracy with respect to constant image size (72 × 72), varying epoch & no of images in the data set. it observed that the accuracy of the proposed algorithm is increased with increase in epochs and no of images in the data set. Fig. 4 we plotted for fixed dataset size, the graph of epochs, and image size against accuracy.

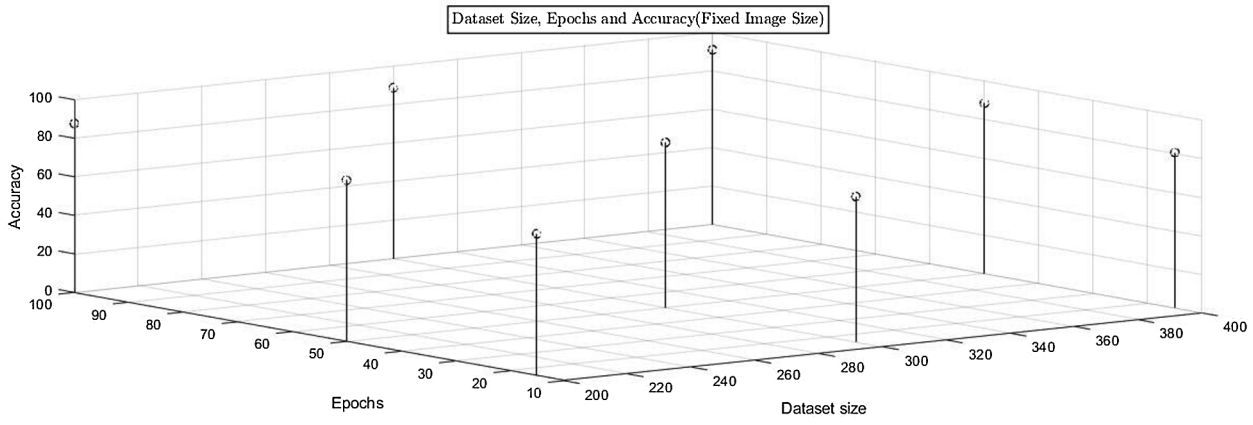

Figure 4: Accuracy of the proposed solution for fixed image size, variable epochs, and variable dataset size

Fig. 5 shows that we get better results with higher image size. In Fig. 5, the proposed method gives, for fixed Epoch = 15, variable image size and variable data size, higher accuracy for higher resolution images, and higher data size with lower missrate.

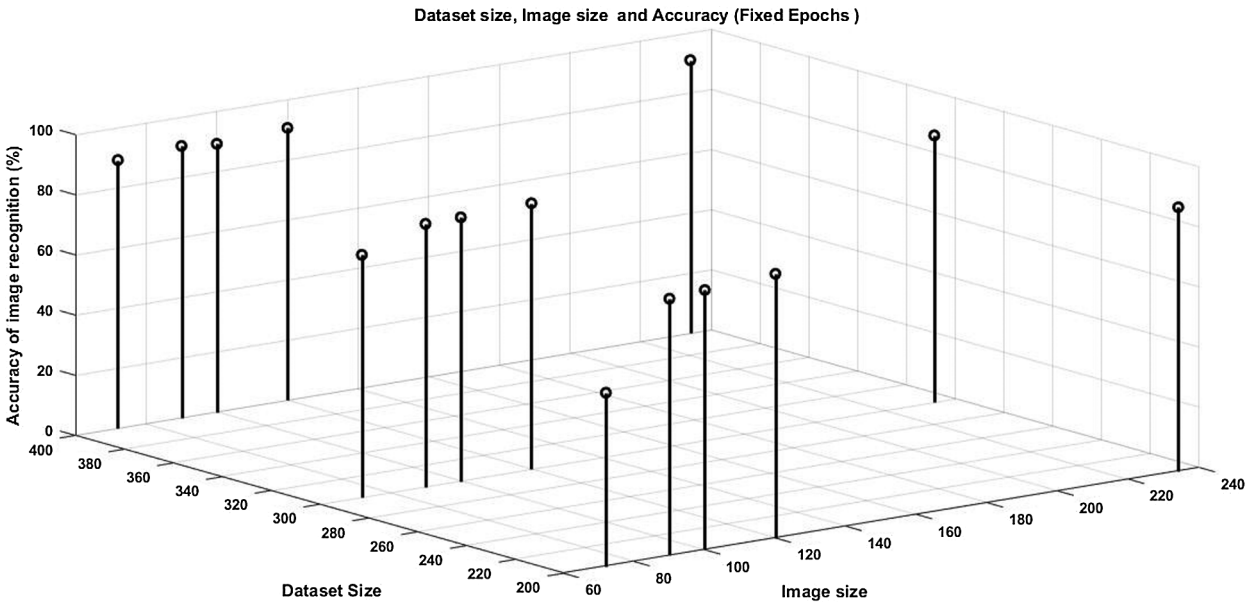

Figure 5: Accuracy of the proposed solution for fixed epoch = 15, variable image size, and variable dataset size

Every recognition system faces 4 major problems: shadow, rotation, scaling, and mirror images. Every recognition system must handle all these problems one by one. However If we train the system on images and If we can use a dataset of appropriate size, all these problems are automatically taken care of by the convolutional neural networks.

5.2 Edit Distance

An Edit Distance is the number of edits needed to convert, a sequence A into another sequence B. The output V from the algorithm "Recognize the Dynamic Gesture" in Section 4, is compared with all the words in the data store, and we choose the string with minimum edit distance.

$$Correct\ Classification\ Rate\ (CCR) = \frac{S_c}{T_s} \times 100$$

where, S_c, T_s represents the total number of samples recognized correctly using Edit Distance & the total number of samples, respectively.

$$Miss\ Classification\ Rate\ (MCR) = \frac{S_{ic}}{T_s} \times 100$$

where S_{ic} represents the total number of samples recognized incorrectly using edit distance. The following is the results from the edit distance algorithm:

Tab. 2 presents results for calculating edit distance for words of length 3, 4, 5, 6, and 7 characters respectively.

5.3 Videos

In PSL gestures are usually 2–5 seconds long. The videos used to form DS are converted to images after passing through the video summarization process. The image labels are stored along with the video label in

the dataset DS. The video recognition process starts with the input of a gesture by the signer. CNN recognizes these images. As the images are quite complicated so learning rate is kept very low and returned labels are stored in the form of a string sequence $s_1 s_2 s_3 \ldots s_n$ at a particular location. Tab. 3 tells us about the impact of increasing epochs on mini-batch accuracy for 200 images per label.

Table 2: Edit distance performance

Word length	Total time taken by all words in DS in (ms)	Time to find minimum in (ms)	CCR%	MCR%
3	800	650	100	0
4	1000	814	99.63	0.37
5	1120	917	99.53	0.47
6	1300	1105	99.88	0.12
7	1410	1228	99.67	.33
Average			99.74	0.26

Table 3: Proposed solution performance w.r.t. epochs and minibatch accuracy & misrate

Epochs	Mini-batch accuracy (%)	Miss rate (%)
1	35.16	64.84
17	67.19	32.81
34	82.81	17.19
50	82.03	17.97
67	92.97	7.03
84	94.53	5.47
100	92.19	7.81
150	93.75	6.25
200	96.09	3.91

Fig. 6 shows the relationship between epochs, mini-batch accuracy, and mini-batch loss. The graph shows that with the increase of the number of epochs, the mini-batch accuracy increases and mini-batch loss decreases. At epochs = 50 the proposed system gives a mini-batch loss of 18% approximately. The same decreases to approximately 4% when epochs become equal to 200.

The results can be improved by changing many factors, including the number of images per label, image resolution, learning rate, filter size and number of filters. As an example to test the input video, the following words were chosen: Cap, skirt, scarf, and gloves. The video title and the summarized image labels are stored in DS. This is done for all the selected videos. For this research, almost 1000 words are selected. However, more words can be added in the data store DS with an increased cost in terms of training time and a little impact on testing time. As some of the gestures are repeated, so a total number of classes, i.e., image labels, do not exceed a limit. Let's now test an input video, V, summarized to frames $f_1 f_2 f_3 \ldots f_n$. The images are sequentially compared the edit distance algorithm is $O(n^2)$ for comparing two strings.

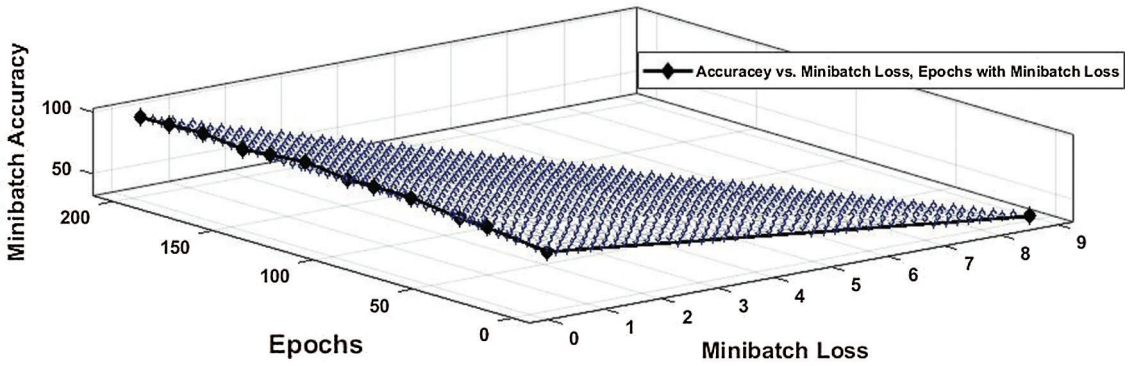

Figure 6: Epochs, minibatch accuracy%, and minibatch loss%

The computational complexity of deep neural networks is determined by matrix multiplication, nonlinear transformation and weight sharing. Dropout helps in keeping the computational complexity in the polynomial-time domain. The training part of the proposed solution is the most time-consuming part. It takes hours to get training results, however, the testing is of the order of a second. The system gives an accuracy of 90.03% on training data. The edit distance algorithm gives an accuracy of 99.99%. For the subset of words selected, it was found to be 99.74%, so the proposed system gives an accuracy of 90.79% on training data. This accuracy can be increased well above 91% by increasing the number of images per class in the dataset, increasing image resolution and increasing number of epochs.

6 Comparison with Existing Techniques

The results of the proposed solution were also compared with other existing techniques. The proposed technique achieves accuracy comparable to those provided by [25–27,29,44]. Tab. 4 shows the comparison of the proposed technique with other techniques. The proposed method performs reasonably well in terms of this metric.

Table 4: Comparison of some existing techniques

Technique name	Accuracy %
Two-Stream Convolutional Networks for Action Recognition in Videos [29]	79.34%
Long-term Temporal Convolutions for Action Recognition [26]	80.5%
Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Videos [25]	86.02%
Convolutional Two-Stream Network Fusion for Video Action Recognition [39]	84.85%
Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks [27]	71.31%
The Proposed Dynamic Gesture Recognition Using CNN and Edit Distance	90.79%

7 Conclusion and Future Work

The research is an effort to facilitate the deaf society and to provide an efficient touch-free interface to users of smart devices. The proposed technique has the edge that it gives good accuracy in a constraint-free environment. The proposed methodology provides a framework for sign language recognition that can be

materialized for any sign language. A larger dataset can also give better video recognition accuracy. A better algorithm for string matching of the combined output of the image recognition algorithm, which gives improved results over edit distance, is left as future work. A detailed complexity analysis of the system has also been left as future work.

Acknowledgement: Thanks to our families & colleagues, who supported us morally.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Kanehira, L. V. Gool, Y. Ushiku and T. Harada, "Aware video summarization," in *IEEE Conf. on Computer Vision and Pattern Recognition*, alt Lake City, UT, USA, pp. 7435–7444, 2018.

[2] S. E. D. Avila, A. P. Lopes, J. A. Luz and A. A. Arajo, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[3] T. Sebastian and J. J. Puthiyidam, "A survey on video summarization techniques," *International Journal Computer Application*, vol. 132, no. 13, pp. 30–32, 2015.

[4] S. Kamoji, R. Mankame, A. Masekar and A. Naik, "Key frame extraction for video summarization using motion activity descriptors," *International Journal of Research in Engineering and Technology*, vol. 62, pp. 291–294, 2014.

[5] B. Gong and K. Grauman, "Diverse sequential subset selection for supervised video summarization," in *Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, pp. 2069–2077, 2014.

[6] E. J. Cahuina and G. C. Chavez, "A new method for static video summarization using local descriptors and video temporal segmentation," in *IEEE Conf. on Graphics, Patterns and Images*, pp. 226–233, 2013.

[7] Y. Shi, H. Yang, M. Gong, X. Liu and Y. A. Xia, "Fast and robust key frame extraction method for video copyright protection," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–14, 2017.

[8] K. Mahmoud, N. Ghanem and M. Ismail, "Vgraph: An effective approach for generating static video summaries," in *IEEE Int. Conf. on Computer Vision Workshops*, Sydney, Australia, pp. 811–818, 2013.

[9] G. Guan, Z. Wang, S. Lu, J. D. Deng and D. D. Feng, "Keypoint-based keyframe selection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 729–734, 2013.

[10] E. Asadi and N. M. Charkari, "Video summarization using fuzzy c-means clustering," in *20th Iranian Conf. on Electrical Engineering*, Karaj, Iran, pp. 690–694, 2012.

[11] Q. Zhang, S. P. Yu, D. S. Zhou and X. P. Wei, "An efficient method of key-frame extraction based on a cluster algorithm," *Journal of Human Kinetics*, vol. 39, no. 1, pp. 5–14, 2013.

[12] M. Danelljan, A. Robinson, F. S. Khan and M. Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *European Conf. on Computer Vision*, Springer, Cham, pp. 472–488, 2016.

[13] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *IEEE Int. Conf. on Computer Vision*, Darling Harbour, Sydney, pp. 3551–3558, 2013.

[14] S. Bansal, S. Bhowmick and P. Paymal, "Fast community detection for dynamic complex networks," in *Complex Networks*, Springer, Exeter, UK, pp. 196–207, 2011.

[15] M. K. Bhuya, D. A. Kumar, K. F. MacDorman and Y. Iwahori, "A novel set of features for continuous hand gesture recognition," *Journal on Multimodal User Interfaces*, vol. 8, no. 4, pp. 333–343, 2014.

[16] M. Ajmal, M. Naseer, F. Ahmad and A. Saleem, "Human motion trajectory analysis based video summarization," in *16th IEEE Int. Conf. on Machine Learning and Applications*, Cancun, Mexico, pp. 550–555, 2017.

[17] S. Mei, G. Guan, Z. Wang, S. Wan, M. He *et al.,* "Video summarization via minimum sparse reconstruction," *Pattern Recognition*, vol. 48, no. 2, pp. 522–533, 2015.

[18] K. Muhammad, T. Hussain and S. W. Baik, "Pt us cr," *Pattern Recognition Letters*, vol. 3, no. 2, pp. 173–178, 2018.

[19] M. Buri, M. Pobar and M. I. Kos, "An overview of action recognition in videos," in *40th Int. Convention on Information and Communication Technology, Electronics and Microelectronics*, Opatija, Croatia, pp. 1098–1103, 2017.

[20] R. Panda, S. K. Kuanar and A. S. Chowdhury, "Scalable video summarization using skeleton graph and random walk," in *22nd Int. Conf. on Pattern Recognition*, Stockholm, Sweden, pp. 3481–3486, 2014.

[21] A. Voulodimos, I. Rallis and N. Doulamis, "Physics-based keyframe selection for human motion summarization," *Multimedia Tools and Applications*, vol. 79, no. 5, pp. 3243–3259, 2018.

[22] K. Zhang, W. Chao and K. Grauman, "Summary transfer: Exemplar-based subset selection for video summarization," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 1059–1067, 2016.

[23] R. Chellappa, J. C. Chen, R. Ranjan, S. Sankaranarayanan, A. Kumar *et al.,* "Towards the design of an end-to-end automated system for image and video-based recognition," in *Information Theory and Applications Workshop*, La Jolla, CA, USA, pp. 1–7, 2016.

[24] J. Singha and R. H. Laskar, "Hand gesture recognition using two-level speed normalization, feature selection and classifier fusion," *Multimedia Systems*, vol. 23, no. 4, pp. 499–514, 2017.

[25] L. Pigou, A. V. D. Oord, S. Dieleman, M. V. Herreweghe and J. Dambre, "Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video," *International Journal of Computer Vision*, vol. 126, no. 2, pp. 30–39, 2018.

[26] G. Varol, I. Laptev and C. Schmid, "Long term temporal convolutions for action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1510–1517, 2018.

[27] Y. G. Jiang, Z. Wu, J. Wang, X. Xue and S. F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 352–364, 2018.

[28] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan *et al.,* "Long term recurrent convolutional networks for visual recognition and description," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2625–2634, 2015.

[29] X. Wang, L. Gao , J. Song, X. Zhen, N. Sebe *et al.*, "Deep appearance and motion learning for egocentric activity recognition," *Neurocomputing*, vol. 275, pp. 438–447, 2018.

[30] T. Wall, L. T. Tran and S. Soejatminah, "Inequalities and agencies in workplace learning experiences: International student perspectives," *Vocations and Learning*, vol. 10, no. 2, pp. 141–156, 2017.

[31] K. S. Thakre, A. M. Rajurkar and R. R. Manthalkar, "Video partitioning and secured key frame extraction of MPEG video," *Procedia Computer Science*, vol. 78, pp. 790–798, 2016.

[32] C. V. Sheena and N. K. Narayana, "Key frame extraction by analysis of histograms of video frames using statistical methods," *Procedia Computer Science*, vol. 70, pp. 36–40, 2015.

[33] Y. H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga *et al.,* "Beyond short snippets: Deep networks for video classification," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 4694–4702, 2015.

[34] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez *et al.,* "Continuous control with deep reinforcement learning," *ArXiv Preprint ArXiv*, 2015.

[35] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 779–788, 2016.

[36] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[37] H. Nam, M. Baek and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," *ArXiv Preprint*, *ArX-iv:1608.07242*, 2016.

[38] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. Torr, "Fully convolutional siamese networks for object tracking," in *European Conf. on Computer Vision*, Springer, Amsterdam, The Netherlands, pp. 850–865, 2016.

[39] C. Feichtenhofer, A. Pinz and A. Zisserman, "Convolutional two stream network fusion for video action recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 1933–1941, 2016.

[40] X. X. Zhu, D. Tuia, L. Mou, G. S. Xia, L. Zhang *et al.,* "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[41] Y. Song, J. Vallmitjana, A. Stent and A. Jaimes, "Tvsum: Summarizing web videos using titles," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 5179–5187, 2015.

[42] A. E. Zaart, "Images thresholding using isodata technique with gamma distribution," *Pattern Recognition and Image Analysis*, vol. 20, no. 1, pp. 29–41, 2010.

[43] S. Saqib and S. Kazmi, "Video summarization for sign languages using the median of entropy of mean frames method," *Entropy*, vol. 20, no. 10, pp. 748–776, 2018.

[44] N. Ejaz, T. B. Tariq and S. W. Baik, "Adaptive key frame extraction for video summarization using an aggregation mechanism," *Journal of Visual Communication and Image Representation*, vol. 23, no. 7, pp. 1031–1040, 2012.

[45] N. Jaouedi, N. Boujnah and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University Computer and Information Sciences*, vol. 32, no. 4, pp. 1–11, 2019.