Оглавление

[1 Proposed work 2](#_Toc163113259)

[1.1 Pre-processing 2](#_Toc163113260)

[1.2 Deep sign recognition architecture 3](#_Toc163113261)

# Proposed work

The proposal has been divided into two sections as the ASL LVD video sequences are pre- processed and then recognized using cascaded 3-D CNNs. Sections 1.1 and 1.2 describes about pre-processing done for better training of cascaded CNNs and cascaded CNNs respectively. The major components of the proposed model are shown in Figure 1.

## Pre-processing

To effectively train the CNNs, some pre-processing has been done. This reduces the chances of CNNs being trained on noising elements resulting in degraded performance. Since pre- processing is only done while training the network so it is a prior expense of time. Below we outline the various pre-processing stages.

* Each video sequence is first converted into several frames, then each frame is processed individually.
* Original color frame is first converted to a gray-scale image. Then unwanted noise and spots in the frame are removed using median filtering.
* The illumination variations in the frame are canceled out using Histogram equalization. To reduce the computation, each frame is resized to 512 × 384 and normalized to [0, 1].
* Each video sequence is then reduced to the size of 25 distinct frames.
* The processed frames are then combined again to form the video sequence for training 3-D CNNs.

The above process generates the processed video sequences having gray-scale frames. The video sequences that are processed were manually trimmed. This ensures only hand gestures and motions to be present in video sequences for training CNNs. As outlined earlier, the works presented to recognize dynamic ASL are less in number as compared to static ASL recognition. Several authors have tried various feature extraction methods followed by the use of different learning techniques like HMMs, Recursive partition tree, and ANMM. But the use of deep learning techniques has not yet been presented. So, we tried to explore the deployment of CNNs to resolve the problem of dynamic ASL recognition.

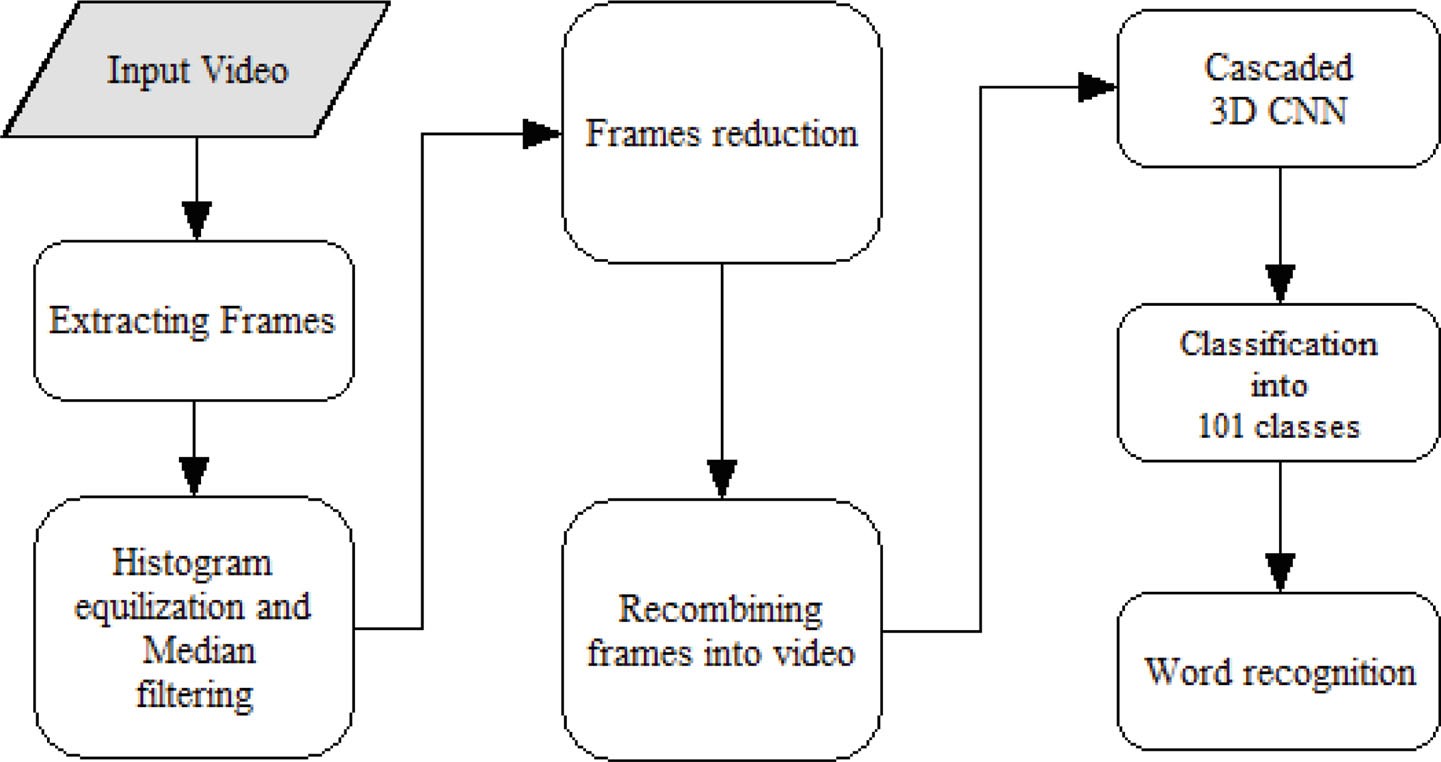


Figure 1 – Various components of the proposed model

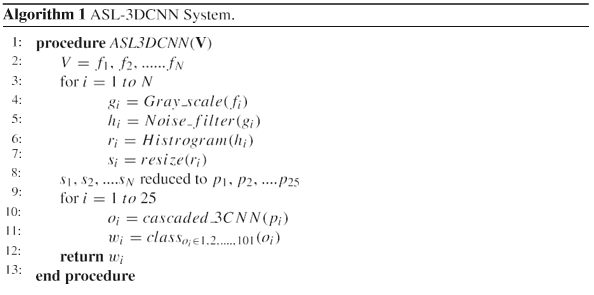


Figure – Algorithm

## Deep sign recognition architecture

The concept of neural networks came into existence by the works, while the concept of deep learning is coined fairly recently in the mid-2000s by Hinton and his collaborators. Figure 3 shows architecture of CNN in proposed method.

As the name suggests, it focuses on the development of a sequence for feature recognition maps, stacking one layer on top of the previous layer, and where each layer recognizes the extended features provided the previous layer, with the final layer performing classification. For example,

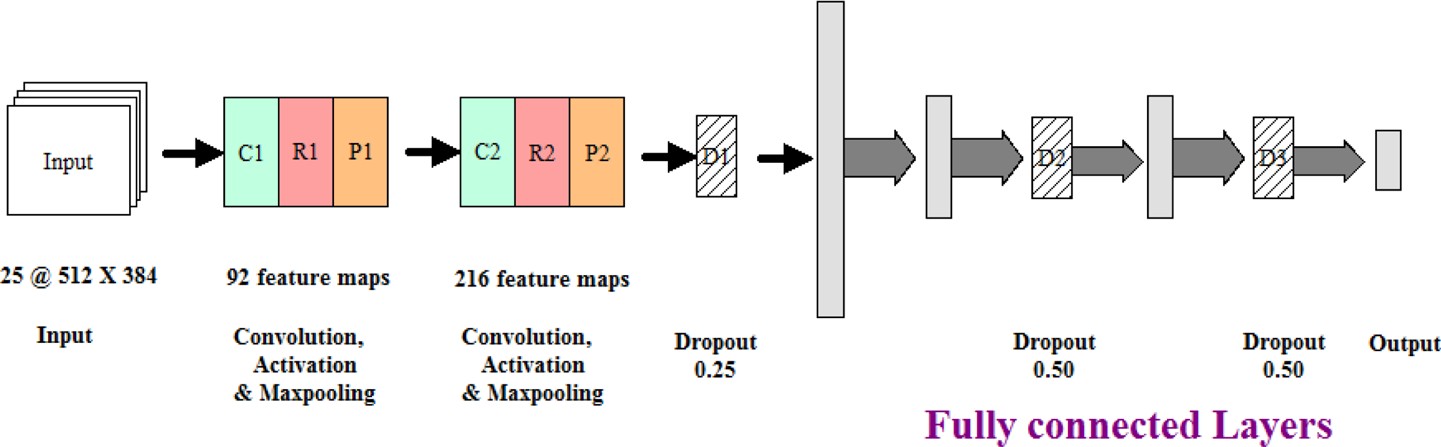


Figure 3 – Work flow of the proposed model

to recognize objects in images, the first layer learns to understand patterns in edges, the second layer combines that pattern of edges to form motifs, the next layer learns to combine motifs to attain patterns in parts, and the final layer learns to recognize objects from the parts identified in the previous layer. The summary of CNN is given in Table 1.

The convolution operation is widely used in the field of image processing. The convolution layers on CNN also work on the same principle. The convolution of an input *x* with kernel *k* is computed by (1), where *x* is an image in the input layer or a feature map in the subsequent layers. The convolution kernel, *k* is a square matrix having dimension specified by the user. The number of feature maps is a hyper-parameter that is determined experimentally. For a 3D kernel, the convolution is defined as in (1).

|  |  |
| --- | --- |
|  | () |

where is the pixel,

is kernal,

and m are 3 Dimensional values,

92 feature maps or filters are used after the input layer. The kernel size defines the receptive field of the hidden neurons in feature maps.

Table 1 – Configuration of 3-D CNN model for Sign recognition

|  |  |
| --- | --- |
| Fully connected | #neurons: 101 |
| Dropout ReLU  Fully connected | Ratio : 0.5  #neurons: 1024 |
| Dropout ReLU  Fully connected | Ratio : 0.5  #neurons: 4096 |
| Dropout  Maxpooling ReLU | Ratio : 0.25  kernel : 2 × 2 × 2, stride : 2 |
| Convolution  Maxpooling ReLU | #filters: 216, kernel : 15 × 15 × 3, stride :1  kernel : 2 × 2 × 2, stride : 2 |
| Convolution  Input | #filters: 92, kernel : 25 × 25 × 6, stride : 1  512 × 384 × 25 gray-scale video |

In the case of the 3D kernel, the last dimension specifies the number of frames falls into the receptive field. It acts as the filter for searching a specific pattern in the input image. The stride defines the movement of the kernel across the input image. Lesser the stride more accurate the feature maps regarding the patterns. In our model, we took the stride of 1 in all the kernel dimensions. We used ReLU (Rectified Linear Unit) as activation function, which enhances the learning process of the network. For the input *x* the output of ReLU is defined as in (2).

|  |  |
| --- | --- |
|  | () |

A smooth approximation to ReLU is the analytic function also called soft plus function is defined as in (3).

|  |  |
| --- | --- |
|  | () |

To avoid the exploding gradient problem, we employed dropout layers having the ratio to be 0.50 & 0.25. Dropout layers neglect the input from some neurons in previous layers. This avoids the exploding as well as the vanishing of the gradient. Moreover, this also avoids the over-fitting of the network while training, promising higher accuracy of test data. A pooling operation is applied to reduce the impact of translations and reduces the number of trainable parameters that would be needed. All the layers discussed above collectively act as a single convolution layer. In our model, we deployed two convolution layer. One having 92 feature maps and the other having 216 feature maps. Moreover, we also changed the kernel size in the next layer for better training and testing.

A fully connected layer could be understood as the feed-forward neural network. The feature maps obtained after both convolution layers act as input to fully connected layers. The last convolution layer contains 216 feature maps with a matrix having 115×78×4 as the dimension is reshaped into a single 7,750,080 dimensional vector. This vector act as input to a three-layer neural net with 4096 nodes in the first hidden layer, 1024 in the second hidden layer, and 101 class nodes, one for each word in the lexicon and last one for *NULL* denoting that the word doesn’t belong to above 100 classes. The softmax layer transforms the output of various neurons into probability.

With time, many robust and fast training algorithms have been presented by many authors. More recently, Kingma et al. proposed a new training algorithm called Adam optimization basically in neural networks for speeding up the learning process. They used the concept of second-order moments and their correction in training. We used an Adam optimization technique for backpropagating the error. It has many benefits over the traditional Stochastic gradient descent method (SGD). It removes the major drawback of SGD viz. slow training. The various parameters in Adam optimization are stepsize , exponential decay rates for moment estimation , objective function and moment vectors .

The gradient concerning at any time instance is defined as in (4).

|  |  |
| --- | --- |
|  | () |

The various salient features of our work are:

1. 3-D CNN’s cascaded is employed as a deep learning framework for recognizing the dynamic hand sign with better accuracy.
2. The proposed work can work for Signer-independent as well as surroundings independent.
3. To consider the different viewpoints, the cascaded CNNs have been deployed for better training.
4. The proposed work outperforms the existing state-of-art models in terms of precision (3.7%), recall (4.3%), and f-measure (3.9%).
5. The computing time (4 milliseconds per frame) of the proposed work shows that the proposal may be used in real-time applications.