Turkish Journal of Computer and Mathematics Education Vol.12 No.9 (2021),1423-1433

Research Article

Transfer-learning analysis for sign language classification models

Rajesh George Rajan[1] , Dr.P. Selvi Rajendran[2]

[1]Research Scholar, Hindustan Institute of Technology and Science, Padur, Chennai-603103, Tamil Nadu
[2]Professor, Hindustan Institute of Technology and Science, Padur, Chennai-603103, Tamil Nadu

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract—Alphabet sign language recognition with high accuracy is a tedious task in computer vision due to several reasons like the lack of sufficient quantities of the annotated dataset, signer variety, continuous signing, etc. The relative scarcity of labelled data in the sign language recognition field has impeded the exploitation of the different models. For the above-mentioned problems, the researchers perform other augmentation techniques. In this paper, we evaluate four different transfer learning approach on three publically available datasets. The various transfer learning models explore the influence of different data augmentation techniques that should increase the performance of the classification model. All the four model makes a comparison with data augmentation and without data augmentation. This study suggested that the transfer learning can improve the classification accuracy.

Keywords— Transfer learning, deep convolutional neural network, data augmentation, American sign language.

1. Introduction

Sign language (SL) is regarded as the most formal and structured type in different categories of gestures, mostly in expression hand/arm gesture taxonomies. Sign language is a crucial communication medium for the deaf and hearing impaired Sign Language involves hands/arms and deals with non-manual signs that convey semantical meaning through facial expressions and body postures. Sign language recognition is a joint field of study, which covers the mixing of patterns, machine vision, linguistic processing, and natural language. It aims to develop different approaches and algorithms to classify the signs and convey meaning. Automatic sign recognition is a difficult interdisciplinary question not yet fully solved. In recent years, various methods have been used to use machine learning strategies for the understanding of sign language. There have been efforts to recognize human gestures by introducing deep learning techniques. In comparison to conventional networks, networks referring to profound learning paradigms interact with biologically driven architectures and algorithms. The creation of deep networks typically occurs layer by layer and relies on more dispersed characteristics in the human visual layer. Here, the features from signs in the initial layer are given into the subsequent layer, merged into the next layer of more specified characteristics [1].

The Sign Language Recognition System (SLRS) aims to transform the Sign Language into accessible forms and provide a simple medium for communicating with people with hearing disability. Usually, the issue of sign language interpretation was categorized into the vision and sensor-based methods. The images are obtained from a device without sensors or gloves, but using vision-based techniques like a camera. This type of vision-based method is simple and easy to use everywhere [2]. An interpretation and identification of signs using images are assisted by various vision-based machine learning, and image processing approaches.

Deep learning has proliferated in recent years. In image classification, natural language processing, CNNs can achieve much more effective results. So in the latest days, the researchers concentrate on deep learning techniques and applications to solve the issues.However, the relative scarcity of annotated data and high computational power graphical processing units are considered the challenges of recognizing sign language. As CNN having high popularity, transfer learning concept becomes widely accepted. The model can be converged more easily with the help of Transfer Learning.

This paper focuses mainly on techniques based on static recognition. The disadvantages of the static gesture recognition system include unstable and inaccurate recognition under abrupt lightning changes and complex circumstances. The different methods that use different transfer learning approaches help eliminate the need for extraction of features and reduce the calculation power required to obtain better sign recognition accuracy. Some of the concept of the paper is taken from [12]. We designed four models: the end-to-end CNN model, the pre-trained fine-tuned CNN model, cross dataset fine-tuned model and Fusion of Fine-Tuned Inception V3 and VGG-16 model. Here we evaluate and compare the alternative strategies to the model using with and without data

augmentation and discuss their outcomes. Data Augmentation has two benefits, such as it helps in preventing overfitting and gives the capability to generate new data from limited data. The following parts of this paper: related works, proposed system and methodologies, results and discussion and conclusion, respectively under sections II, III, IV and V.

2. Releated works

A brief overview of different articles covering various methods used in SLRS is given in this section.

Munib et al. [4] introduced a classification of static words dependent on Hough transform in American Sign Language. 300 specimens of 20 sign images were obtained, and the reference origins, orthogonal variables and shapes were collected. These experimental results have shown that the suggested approach is resilient to changes in sign size, position and direction.

A recognition scheme based on 24 static American Sign Language (ASL) alphabet signs using colour and depth representations was introduced by Pugeault and Bowden [5]. Gabor filters are employed to obtain characteristics at various scales, and the multiclass random forest has been used as the classifier. In the leave-one-out trial, they achieved a 49 per cent recognition rate. They also created a dataset called American Fingerspelling Dataset, and in this type of sign language analysis, it is the most widely used benchmarked dataset. Karayilan and Kiliç [7] proposed a neural network-based sign-language detection framework. They obtained signs with a camera that extracted raw and histogram attributes. For raw and histogram characteristics, an average accuracy of 70% and 85% were achieved, respectively.

A framework for identifying single-handed static American sign images obtained using the camera was proposed by Ragab et al.[13]. To remove features, they adopted the Hilbert space-filling curve technique because it helps maintain the pixels' localization property and contributes to better performance in representing shapes with uniform backgrounds.

A static alphabet recognition CNN model for the Indian Sign Language (ISL) method was introduced by Sruthi C. J and Lijiya A [3]. For static ISL alphabet recognition, they introduced the binary silhouette of the signing hand region. The built model achieved an accuracy of 98.64 per cent. Two custom CNN based models have been developed by P.Paul et al.,[11] that can identify 24 static ASL signs. They partition one dataset into colour images, and a mixture of colour and depth images is required for the other package. They achieved 86.52 per cent and 85.88 per cent precision on both their models and compared their outcome with these two sets of the dataset using transfer learning with pre-trained models such as VGG 19, VGG 16. Zamani and Kanan[14] developed a camera-based system for recognizing American alphabets and numerals. Two thousand five hundred twenty single-handed static sign images were obtained, and 99.88 per cent accuracy was obtained.

Kumar et al.[15, 16] designed a system to recognize sign language in American Sign Language to identify both static and dynamic signs. They used Zernike moments to define hand orientation in static signs and track a fingertip's centre of gravity for dynamic signs.

Numerous machine learning algorithms have been proposed to help the automatic detection of sign language recognition. Deep Neural Networks (DNNs) are enforced to various benchmark data sets among these machine learning algorithms. Transfer learning in the machine learning context implies utilizing the outputs of many DNN applications. By assigning 4 transfer learning techniques to deep learning models, we test all these classification models, including the baseline model (End-to-End CNN model). The first model (baseline approach) is a Convolutional Neural Networks (CNN) end-to-end type separately tested on all datasets. We have fine-tuned the pre-trained CNN models like inceptionv3 and vgg-16, that have previously been trained in the ImageNet database for the second type (fine-tuning). For the other model, cross dataset fine tuning, it is similar to leave one out procedure. N-1 dataset is given initially for training and remaining one is given for testing. For the fourth design, features have been extracted from pre-trained networks, and these features are fused with the help of serial based fusion and the feature dimensionality is reduced by PCA and then classify the signs.

To produce better outcomes in a visual recognition task, we use of Transfer learning and CNN. Comparing various deep learning approaches, the fusion of fine-tuned pre-trained model is more effective than an end-to-end strategy. The explanation for this is that a competent CNN model needs massive data and the fusing of both network and appropriate feature selection for classification. Most of the research dealing with any one of the datasets. In order to monitor the efficiency of various transfer learning methods and deep learning, we applied different datasets of different sizes.

3. Proposed system and methodologies

In the proposed framework, data augmentation procedure is enforced to increase the size of the dataset. Moreover, the images without data augmentation and with data augmentation are provided to the proposed deep learning prototype. For this procedure, the deep features of the vector are obtained using the ASLNET model. Subsequently, the deep features are classified using a softmax classifier.

A. Convolutional Neural Networks

In image classification and pattern recognition, one of the inevitable methods is CNN, developed by Lecun et.al., [4]. In CNN, the kernel which contains the parameters pass through the input images and obtaining features of images. During convolution, the filter multiplies the filter's values with the original pixel values of the image. After the filter moves through all the parts of the image, the feature map or activation map has generated The size of the feature maps can be reduced with the help of an operation called pooling and given as input for the following convolution. This method continuous until profound features are extracted. The dense or fully-connected layer(FC) ends up with a SoftMax function for classification. Thus a CNN includes various layers like convolutional layer, fully connected layers, pooling layer and activation layers. Convolution operations are used for extracting the features of the framework, whereas a fully connected network is a classifier for most of these features.

B. Major components of Convolutional Neural Networks

The significant operations and layers involved in the construction of a CNN as mentioned below

1. Convolution operation

It is the most critical block and computational part of CNN. Its parameters consist of a set of learnable kernels. Convolutional operations are done on an image of the size I, with a kernel size of F, weights of W, a stride of size S and P as padding to generate an output of size

$$\frac{(I-F+2P)}{S+1}\times\frac{(W-K+2P)}{S+1} \quad (1)$$

The filter used for feature extraction convolved within the image and produces the feature map or activation map. The output 'O' achieved from mathematical convolutional operation between matrix M of size (P, Q) and matrix N of size (R, S) can be indicate as

$$O(i,j)=\sum_{p=0}^{P-1}\sum_{q=0}^{Q-1}M(p,q)\times N(i-p,j-q) \quad (2)$$

Where $0\le i\le P+R-1$ and $0\le j\le Q+S-1$

1. Pooling

Pooling is used to decrease the image's dimension, i.e. the parameters and computation in the networks, thereby reducing overfitting. In other words, the pooling layers are used to achieve spatial invariance by decreasing the resolution of feature maps but retaining the detected features. The pooling layer does not have any of the learnable parameters, but the stride, padding, etc. are the pooling operation's hyperparameters. [12]. This pooling can be varying in size, and it may overlap.

There are different types of pooling operations: L2 pooling, max and average pooling. Whereas average pooling takes the average of the input values, max-pooling takes the maximum value throughout the layer, and L2 pooling calculates the inputs' L2 norm. The most significant benefits of the pooling operations are the decrease of the image size and the extraction of the visual features separately on the image [5]. We use the max pooling operation over the respective field of (r1, r2) with a stride of 'x' pixels

$$y=\max_i(\{a_i\}) \quad (3)$$

where {ai} set of all ai is an element of (r1, r2) receptive field, where the corresponding field is described as the region in the input space that has a specific CNN's feature.

2. Regularization Techniques

An essential part of the machine learning method is to avoid the overfitting problem. This technique makes a minor modification of the hyper parameter called learning rate, thereby enhancing the model's performance on the unseen data. Various regularization techniques like data augmentation, L1 & L2 regularization, early stopping, dropout, and drop connect are commonly used. For this model dropout, data augmentation and early stopping techniques are used.

The hidden and visible neurons are dropping out spontaneously to handle the overfitting problem. This approach was implemented by Srivastava et al. [6]. The neurons that are dropped out do not grant anything to the backward as well as a forward pass. Thus, neural network samples a distinct architecture at each forward-backwards propagation step, but all of these dropout layers even distribute parameters [8]. Early stopping is a cross-validation strategy to maintain one part of the training as the validation set. When the output of the validation set gets worse, the training on the model is stopped instantly. Early stopping is generally used to control the training when the loss begins to increase (or in other words validation accuracy starts to decrease)

3. Fully connected layers(FC)

After the convolution and pooling layer operations, the data is converted into a single-dimensional vector that acts as the FC layer's input. It includes more than one hidden layer, and the weights are multiplied by the previous layers by the neuron input, and a bias value is added. Via the activation function, the measured value is passed and then moved to the next layer. The calculation for the FC layer is

$$D_f = f(b + \sum_{q=1}^{M} w_{1,q} \times o_q) \qquad (4)$$

where ' 'w' is the weight vector, f'is the activation function, 'O' is an input vector of the q th neuron and 'b' is the bias value.

4. Non-Linearity layers

There are different activation functions like rectified linear unit (ReLU), tanh, sigmoid, and its variants (Noisy ReLU, leaky ReLU, Exponential linear unit (ELU)). Typically, researchers use linear unit functions. Here leaky ReLU function is used. The issue we are faced with using the ReLU is the mapping of negative values in the graph. i.e., the negative values became changed to zero. So this problem is avoided by leaky ReLU so that it compresses the hostile part and allow for non-zero gradient when the unit is not active. The mathematical expression of the activation functions as

$$f(x) = \max(0, x) \qquad (5)$$

$$f(x) = \begin{cases} x, x > 0 \\ 0.01x, otherwise \end{cases} \qquad (6)$$

$$f(x) = \begin{cases} x, x \geq 0 \\ a(e^x - 1), otherwise \end{cases} \qquad (7)$$

Here these equations are simple ReLU,(5) leaky ReLU(6), and exponential linear unit (7) where 'a' is used as a hyper parameter and $a \geq 0$.

C. Image Data Augmentation based on image manipulations

A vast amount of labelled data is needed to train a proper convolutional neural network with the best performance. The annotating process is a tedious task and costly process. The conventional method implies various transformation techniques and introduces multiple variations in the images and keeps all the recognizable features. This section describes the geometric augmentation techniques.

a. Flipping: Flipping an image means reversing the images across the horizontal or vertical axis and obtaining a mirror image. This method is easy to implement. The vertical flip is also equivalent to rotating the images by 180^0.

b. Rescale: The image can be either inward or outward. It resizes the images according to a factor called a scaling factor(SF). i.e. reconstruction of the image according to SF.

c. Rotation: This augmentation technique is achieved by revolving the image on an axis between 1 ° and 25 °, either left or right.

d. Crop: Apart from scaling these operations, random cropping is done randomly selecting a sample from the original image.

e. Colour augmentation: In this augmentation, altering an image's colour properties by changing its pixel values. This augmentation consists of brightness, contrast, saturation, and hue operations. In this case, the brightness operation is performed, and the resultant image becomes either darker or brighter compared to the original input.

f. Shear: It is a bounding box transformation with the help of a transformation matrix. Shear can be done either horizontally or vertically with a shearing factor.

Training data set	
Parameters	**Arguments**
Shear Range	.2 degree
Rescale	1./255
Rotation	25 degree
Horizontal Flip	True
Center- cropped	True
Height Shift factor	0.1
Width Shift factor	0.1

Table 1. Augmentation techniques applied to Dataset

D. Different deep learning architectures

1. End-to End ASLNET model architecture

This study focuses on the issue regarding the scarcity of annotated data and hence facing the overfitting. This issue is fixed using image augmentation and a deep CNN-based neural network to create a model called ASLNET to classify the signs. All these operations are done with the help of tensor flow and Keras [9] [10]. Fig.1 shows the general schema of end-to-end model. The method is based on a supervised model, and the whole portion is split into training and testing for CNN. Fig.2 displays the diagrammatic presentation of the model ASLNET. The image input size is 64x64 with 32 different filters, and the size of the first kernel is (5,5). Then 64 output filters are generated in the second layer kernel size (3,3) and then continuously provide the pool size (2,2) for all the pooling layer, and use 25 per cent of dropout in the convolution in the first drop out layer, followed by 35 per cent of drop out in the dense layers. For modification and normalization of the previous layers' activation values, batch normalization layers are used. ReLu layers operate the input threshold to remove the noisy and dark field effects. The role of max-pooling layers is to decrease computational complexity by the input dimensions. The filter uses the Glorot normal initializer, also called the Xavier normal initializer, to initialize the initial random weights. The limit value is sort (2 / (number in + number out)), and the random values are pulled from the Gaussian distribution instead of the uniform distribution: The traditional fixing of weights is difficult because there's no right way to determine the initialization limit values and also, different peoples use different values, and it may not be documented.

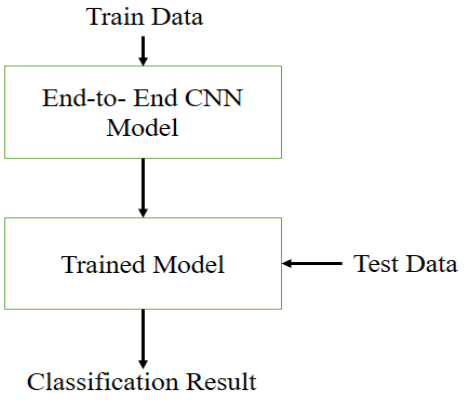

Fig:1 End-to-End Model

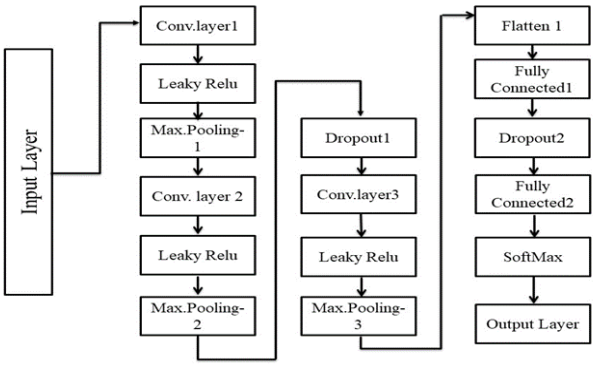

Fig:2 ASLNET CNN model

2. Transfer learning methods

Machine learning models are mainly designed to function independently and must be restored as modifications are needed to data and features. However, for specific tasks, the previous information gained in machine learning can also often be used. Instead of fixing a model that usually takes up much attempt, transfer learning is aimed to restate the prototype so that it can significantly reducing the time of model creation and improving the model's efficiency. Different learning methods, such as transfer learning approaches, focus on domains, transfer learning from pre-trained models and feature spaces.

a. Fine Tuning

When not a sufficient number of training data are available in a domain, training from scratch is not suitable for Convolutional Neural Networks. For these type of problem, a pre trained model is used so that it is always trained by huge dataset. Here we use VGG-16 and Inception V3 model for fine tuning. These basic model architecture is shown in Fig.3 and Fig.5. These CNN models are well trained with ImageNet image database. Both these models have 1000 classes. In fine tuning concept, last three layers namely softmax, fully connected, and classification layer are removed and instead of the 1000 classes, we need to replace with the dataset classes. Finally, the current network architecture is trained with the new sign language data sets. The data sets are fine-tuned with Inception V3 and VGG-16 separately.

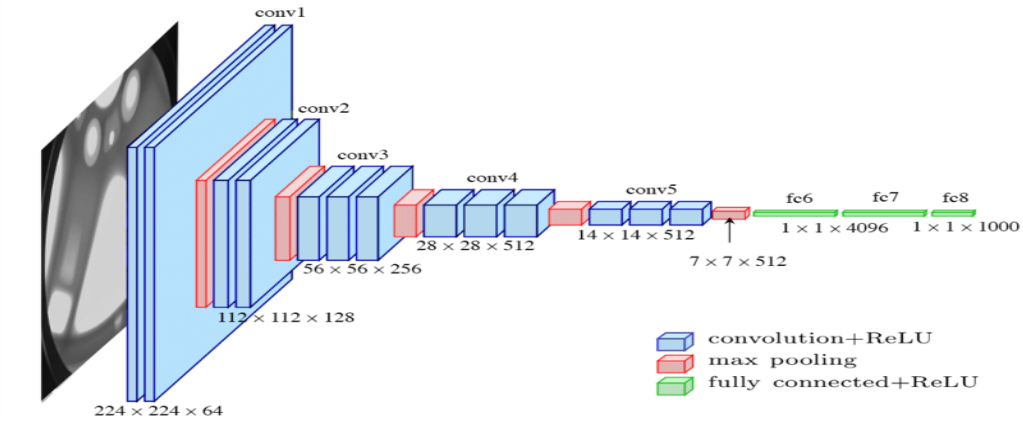

Fig:3 Basic architecture of VGG-16

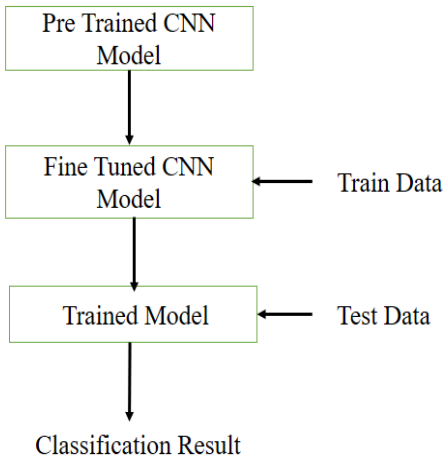

Fig:4 Fine-Tuning Model

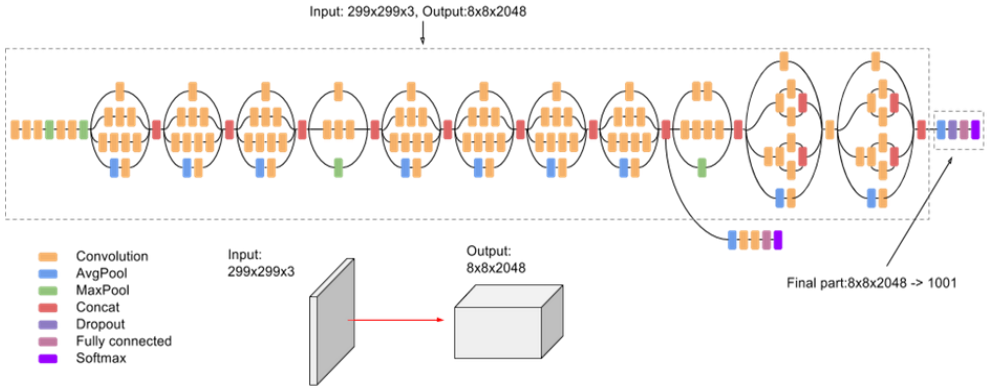

Fig: 5 Basic architecture of Inception V3

b. Cross Dataset Fine-Tuned Model

Here in this model we are given all the dataset for the End-to-End CNN model for training. A mixture of datasets is used in this prototype to train the CNN model from scratch. One data set is omitted from the training sets and is used for testing during this process. These three datasets are being used once for testing. The commonly used term for transfer learning is fine tuning. i.e., this is similar to leave one out strategy so that one set of the dataset is rule out from the training and given to testing. So the (N-1) dataset is given to the end-to-end CNN for training purpose and finally got a trained model. Every dataset is also used for the testing. Through the removal of the last three layers and the inclusion of the new layers, the tuning process is completed. These layers are fully connected, with classification layer and softmax layer. The fully connected layer is modified to the output class number.

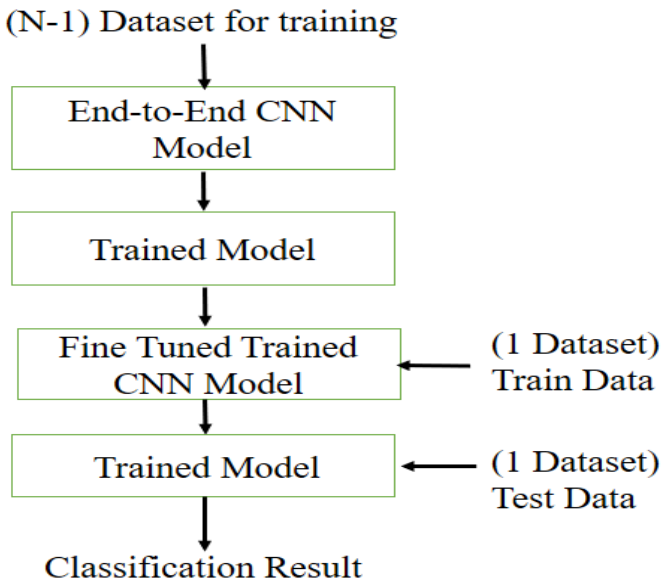

Fig:6 Cross-Dataset Fine-Tuning

c. Fusion of Fine-Tuned Inception V3 and VGG-16 architectures

In this type of network architecture, the Inception-V3 and VGG-16 networks are given directly with input images from the dataset. The two pre-trained models were fine-tuned by modifying the output layer class, and the images were re-sized to accommodate the models' input layers. Activation values from the pre-trained CNNs or FC layers are obtained in the same way as in the previous process. After getting the features from the inception and VGG models, both the features are fused together as serial-based feature fusion. So here we are getting the total of features from both the pre-trained models. A PCA-based feature reduction is performed from that total feature vector so that the appropriate features only given to the classifier for classification.

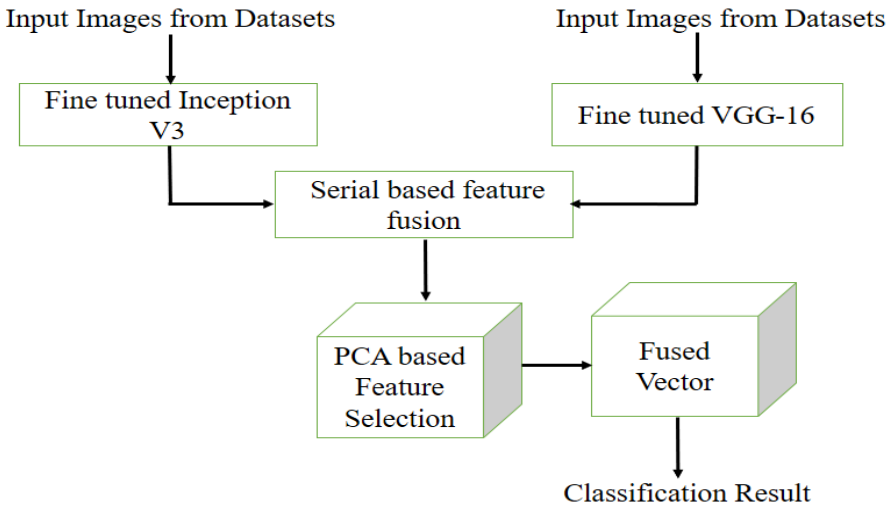

Fig: 7 Fusion of Fine-Tuned Inception V3 and VGG-16 architectures

4. Results and discussion

We explain the data sets in this section first and then give our results and our analysis.

1. Kaggle Dataset

American Sign Language Letter database called Kaggle is used for recognition. It consists of hand gestures of alphabets except 'J' and 'Z' as do not have static postures and three additional signs ('space', 'delete', and

'nothing'). These three additional signs are avoided for this experiment. A total of 87,000 images and 3000 images per class is available in the dataset. The dataset contains the color images of 200 X 200 pixels. A sample dataset depicted in Fig.8. To build a good model for image classification, need an enormous dataset.

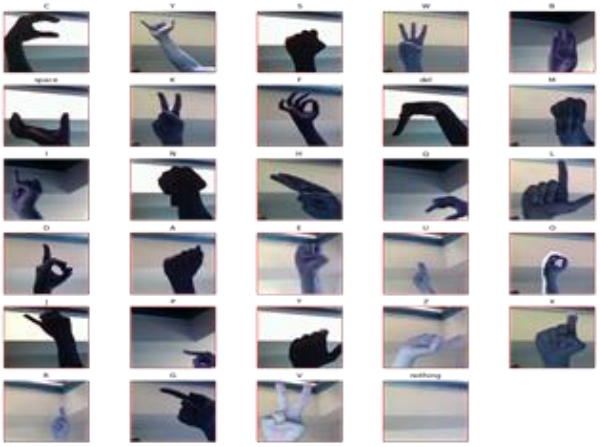

Fig 8: Kaggle fingerspelling dataset

2. ASL Finger Spelling Dataset

The ASL Finger Spelling benchmark dataset is the dataset used in this proposed work. It contains both colour and depth images collected from five different users and contains twenty-four static signs, except the letters j and z, as both temporal and spatial relationships are needed for these two letters. Fig.9 provides a subset of the dataset. There are 95,697 images in total, and approximately 4000 images are contained in each alphabet for each user. With the help of Kinect, five people with non-identical lighting conditions and background conditions, record the ASL data set. About ~500 non-identical hand gesture images are present in the ASL dataset. This dataset is also challenging since it has different backgrounds and different illumination.

Fig 9: ASL fingerspelling dataset [5]

3. Massey Dataset

It is American sign language dataset consists of 2425 color images of 36 classes. It was released in 2011 by the Massey University. This database also includes difference in illumination, scale, and rotation. All the images are in RGB color modes with black background. The sample is mentioned in Fig.10. In this dataset, 36 different groups were treated: 26 for alphabets (a-z) and ten for numbers, respectively (0-9). Here in the dataset, we avoid the numbers also for the unification of all datasets. In this dataset, the images are not in a similar format regarding the size and shape, so the initial step is to resize the image in to similar size.

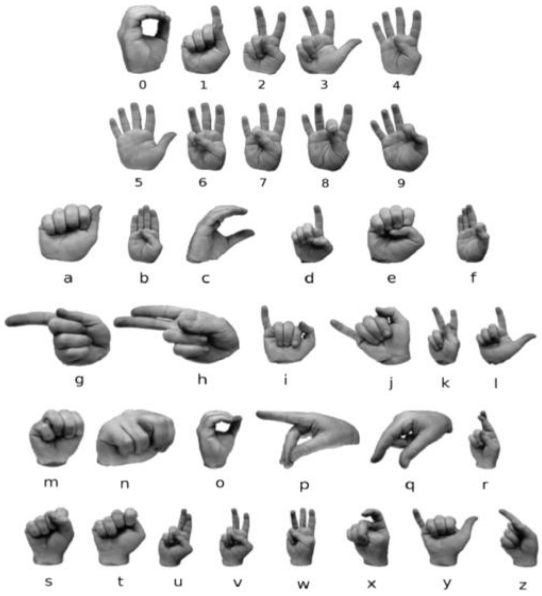

Fig 10: Massey dataset [17]

4. Results

In this different types of transfer learning strategies, we divided the total dataset in to 70% for training and 30% for testing. In the deep learning model training, if we are giving the large number of data for training, then obviously the model shows perfect accuracy. For the training phase of all these models, 50 epochs were used. Table 2 summarizes the findings of the CNN end-to-end model. As noted above, this table shows that with higher data samples, the same CNN model works better., i.e., with augmentation, and this finding illustrates the general concept of deep learning models that imply better classification outcomes for a model trained with a massive amount of data.

Dataset	Accuracy without Augmentation	Accuracy with Augmentation
1. Kaggle	93.36	95.79
2. ASL Fingerspelling	88.13	91.35
3. Massey	92.20	96.08

Table 2: Accuracy of end to end CNN model

Table 3 offers the findings of the cross-dataset fine-tuning analysis. The result of the Massey sign language dataset is increased by approximately three per cent, but the outcome on other datasets is close to the end-to-end results of CNN. This experiment shows that increasing the training data size emphatically influences the model's output on datasets with a smaller sample size. Also, the Massey dataset is arranged in such a way that it was having the segmented images. So the training is much better for the system to learn the pattern.

Dataset	Accuracy without Augmentation	Accuracy with Augmentation
1. Kaggle	94.19	98.84
2. ASL Fingerspelling	90.66	92.87
3. Massey	96.39	98.59

Table 3: Accuracy of cross dataset fine tuning

The outcomes of the pre-trained models' fine-tuning are shown in Table 4. The pre-trained models are trained by millions of images from the ImageNet dataset. Fine-tuning these models, exceptionally though models are trained with images from various contexts, produce more outstanding performance than end-to-end models on precisely limited-element datasets. According to this experimental study, it is noted that the efficiency of the Inception V3 network is comparatively better than that of the VGG-16, but the gap between the two methods is not very significant.

Dataset	Accuracy without Augmentation	Accuracy with Augmentation

1.	Kaggle	94.27	98.83
2.	ASL Fingerspelling	86.59	92.26
3.	Massey	96.43	99.16

Table 4: Accuracy of pretrained model fine tuning

In the fusion of fine-tuned pretrained models like VGG-16 and inception V3 networks, the accuracy is shown in Table 5. Compared to all the previous models, these model obtained the highest accuracy. From the previous tables, we make a consensus that if the system has a large number of data's and if the patterns quickly found out by the neural network, it becomes a better model in the area of performance.

Dataset		Accuracy without Augmentation	Accuracy with Augmentation
1.	Kaggle	94.86	99.02
2.	ASL Fingerspelling	90.89	95.82
3.	Massey	96.03	99.63

Table 5: Accuracy of fusion of fine-tuned pretrained model

5. Conclusion

The SLR problem is a wide research area and now a day the revolution of the artificial intelligence and computer vision helps to overcome all the issues up to an extent. This research showed that transfer learning enhances the effectiveness of deep learning models and, in particular, models that apply deep features and use fine-tuning with data augmentation deliver higher classification with respect to another techniques of transfer learning. This finding suggests that other transfer learning methods should also be taken into account in the case of low precision, rather than implementing an end-to-end CNN model for sign language recognition. In this paper, an end to end and different types of transfer learning models based on neural network in order to classify the signs from the Kaggle, ASL fingerspelling and Massey dataset images. Here we focused on only the static alphabets from these datasets. The machine was trained on all the alphabets and obtained different accuracies with augmentation and without augmentation.

References

A. Oyedotun OK, Khashman A, Deep learning in vision based static hand gesture recognition. Neural Comput Appl 28(12):3941–3951 (2017)
B. Rajesh George Rajan and M Judith Leo, "A Comprehensive Analysis on Sign Language Recognition System", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6, March 2019.
C. Sruthi C J, Lijiya A. "Signet: A Deep Learning based Indian Sign Language Recognition System" , 2019 International Conference on Communication and Signal Processing (ICCSP), 2019.
D. Munib Q, Habeeb M, Takruri B, Al-Malik HA (2007) American sign language (ASL) recognition based on Hough transform and neural networks. Expert Syst Appl 32(1):24–37.
E. Pugeault, N., Bowden, R., 2011. "Spelling it out: Real-time asl fingerspelling recognition". In: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, pp. 1114–1119
F. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). "Dropout: A simple way to prevent neural networks from overfitting". The Journal of Machine Learning Research, 15(1), 1929–1958.
G. Karayılan T, Kılıç Ö (2017) Sign language recognition. In: IEEE international conference on computer science and engineering (UBMK), pp 1122–1126.
H. Jain, R., Jain, N., Aggarwal, A., Jude Hemanth, D.," Convolutional Neural Network based Alzheimer's Disease Classification from Magnetic Resonance Brain Images", Cognitive Systems Research (2018), doi: https://doi.org/10.1016/j.cogsys.2018.12.015.
I. F. Chollet, "Keras," 2015. [Online]. Available: https://keras.io/.
J. M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2016.
K. Pias Paul, Moh. Anwar-Ul-Azim Bhuiya, Md. Ayat Ullah, Molla Nazmus Saqib, Nabeel Mohammed, and Sifat Momen. "A Modern Approach for Sign Language Interpretation Using Convolutional Neural Network", 16th Pacific Rim International Conference on Artificial Intelligence Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Lecture Notes in Artificial Intelligence, Proceedings, Part III

L. Kaya, A.; Keceli, A.S.; Catal, C.; Yalic, H.Y.; Temucin, H.; Tekinerdo ˘gan, B. Analysis of transfer learning for deep neural network based plant classification models. Comput. Electron. Agric. 2019, 158, 20–29.

M. Ragab A, Ahmed M, Chau SC (2013) Sign language recognition using Hilbert curve features. In: International conference image analysis and recognition. Springer, Berlin, Heidelberg, pp 143–151

N. Zamani M, Kanan HR (2014) Saliency based alphabet and numbers of American Sign Language recognition using linear feature extraction. In: 4th IEEE International eConference on computer and knowledge engineering (ICCKE), pp 398–403.

O. Kumar A, Thankachan K, Dominic MM (2016) Sign language recognition. In: 3rd IEEE international conference on recent advances in information technology (RAIT), pp 422–428.

P. Kumar DA, Kishore PVV, Sastry ASCS, Swamy PRG (2016) Selfie continuous sign language recognition using neural network. In: IEEE annual India conference (INDICON), pp 1–6.

Q. Barczak, A. L. C., N. H. Reyes, M. Abastillas, A. Piccio, and T. Susnjak. , A new 2D static hand gesture colour image dataset for asl gestures. (2011).