Journal of Multi Disciplinary Engineering Technologies
Volume 18, Issue 01

Sign Language Recognition using Transfer Learning: Performance Comparison of GoogleNet, ResNet, and VGG

Supriya Bajpai[1] and Nimish Dharamshi[2*]

[1]IITB-Monash Research Academy, IIT Bombay & Monash University, Powai, Mumbai, 400076, Maharashtra, India.
[2*]Department of Electrical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai, 400076, Maharashtra, India.

*Corresponding author(s). E-mail(s): nimishdharamshi6172@gmail.com;
Contributing authors: supriyamishra39@gmail.com;

Abstract

Sign language recognition is a breakthrough for the less privileged community as it eliminates the need of an interpreter whose use is restricted due to high costs and limited availability. It consists of detection of temporal and spatial configurations simultaneously. In order to support integration of deaf people into the society and to help them lead an independent lifestyle, this paper focuses on the technical aspects to recognize and classify the various hand gestures for their easy identification. We use transfer learning on Convolutional Neural Networks (CNN) models- GoogleNet, ResNet and VGG for the identification of hand gestures. These networks are implemented on TensorFlow framework using Python. The approach is to fine tune a pre-trained network keeping robustness and efficiency in mind. Experiments are conducted on Massey University's dataset and accuracy is used as a metric for performance measure. The results and analysis shows that GoogleNet performed better than VGG and ResNet.

Keywords: Convolutional Neural Network, GoogleNet, ResNet, Sign language recognition, VGG

1 Introduction

Sign language has a huge social impact due to the communication barrier between the physically disabled community like the deaf and the dumb and ordinary people. The focus of sign language recognition (SLR) is to convert the sign language in written or spoken form so as to facilitate communication. It recognizes the human emotions made with the help of fingers, head, hand, arms and face. However, this process is complicated by the fact that there are varying types of gestures and there is no internationally accepted sign language. The purpose of this paper is to automate the process of sign language recognition and to keep account of the accuracy obtained in the process. This paper is divided into seven sections. In section 2 CNN and its different models are summarized. Section 3 discusses transfer learning. In section 4 literature survey of various algorithms used for sign language recognition is done. Section 5 provides the experimental setup details for training various

Published on: November, 2024

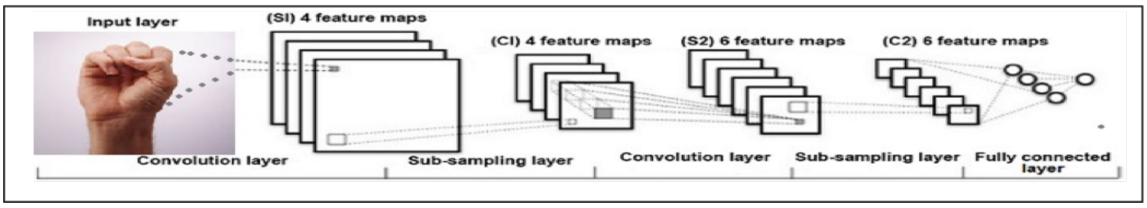

Fig. 1 Convolutional Neural Network Architecture [2]

models. In section 6 we analyze and discuss the results obtained for the classification performance of these models. Section 7 concludes and presents the future scope.

2 Convolutional Neural Networks (CNN)

Artificial Neural Networks are used for solving wide range of classification and prediction problems and for scaling of applications that require large amount of data. CNN is a variant of Neural Networks and is mostly desirable for image classification. It helps in pointing out patterns and features based on input data. It is inspired from a region in our brain called visual cortex. Visual cortex consists of various cells that respond in the presence of certain regions of the visual field. According to an experiment conducted by Hubel and Wiesel [1, 2], some neurons in our brain fired when they detected the presence of edges with a particular orientation. This characteristic of specialized components used for detecting certain features forms the basis of CNN.

The architecture consists of Convolutional layer for extracting features from input, Pooling/Sub sampling layer for reducing the dimensions of input image, non linear layers and the fully connected layer which sums up all the previous layers to determine the target result as shown in Figure 1. The convolutional layer consists of filters or kernels that convolve across the height and width of the input image multiplying their weights with pixel values of the image to form an activation map. Every filter is used for identifying a particular low-level feature (features can be curves, edges, colors etc) in the input image. The filter enables the network to learn to activate itself in the presence of that feature observed at some spatial arrangement. As the number of filters increase along with the depth of the activation map, so does our knowledge about the image. The output of the first layer which is an activation map is fed as an input to the next layer and so on. As the input is passed through more convolutional layers, more complex features are identified. Non linear layers are applied after each convolutional layer to introduce non-linearity in the network. According to researchers, ReLU (Rectified Linear Units) layers performs better than Tanh and Sigmoid functions as the network trains faster without compromising the efficiency [3]. Pooling layer also known as down sampling may be inserted after non-linear layer. The most popular function to apply pooling is max-pooling [4]. Average Pooling and $L-2$ norm pooling are some other examples that can be used in pooling layers [5]. Since the relative location of various features are more important than the specific location of a feature, this layer reduces the spatial dimension of the input by applying a filter across the input and the maximum number in every sub-region is taken as an output. This reduces the computational costs as well as controls over fitting. The fully connected layer is the last layer in the network which takes the output of its preceding layer as its input and predicts the probability of each class and outputs a N dimensional vector where N is the number of classes. Hyper parameters such as stride (how much a filter shifts across the image), padding (input vector is padded with zeroes to preserve the dimensions of the output) controls the size of the output. Some of the models of CNN are discussed in the following subsection.

2.1 VGG

VGG is known for its simplicity and depth due to significant improvements in its architecture over Alexnet. Multiple 3×3 filters are used instead of the large sized kernel filters. It has an error rate of 7.3%. The basic architecture of VGG consists of 19 layers CNN with 2×2 max-pooling layers. VGG enables the learning of more complex features at a lower cost due to the increased network depth. 3×3S layers have an effective receptive field. It basically consists of subsequent convolutional layers followed by pooling layers. VGG is characterized by pyramidal shape as the bottom layers which are closer to the images are wide as compared to the top layers which are deep [6]. VGG model is more efficient on pre trained networks as it takes too long to train if trained from scratch. One major benefit of VGG is the decrease in the number of parameters. The idea behind shrinking spatial dimensions is enforced by doubling the number of filters after each max pooling layer. VGG reinforced the notion that CNN needs to have a deep network of layers so that the hierarchical representation of visual network works.

2.2 GoogleNet

It is a deep convolutional neural network designed by google featuring the inception architecture. Instead of sequential work parallel work is preferred in this network. It has 22 layers and 9 inception modules leading to increased accuracy and performance. It has an error rate of 6.7%. It is based on the idea of sparse connections hence not every output channel is associated with an input channel. It is different from other models because it can either convolve or pool the input directly. The main advantage is that the computational requirements decrease by reduction in data dimensionality. GoogleNet is the diversion from the approach of simply stacking convolutional and pooling layers on top of each other and adding large number of filters. It is capable of extracting volumes of fine grained information due to the availability of network in network layer [7]. Another salient feature of GoogleNet is the inclusion of bottleneck layer leading to massive reductions in computational requirements.

2.3 ResNet

Residual networks or Resnet is a network architecture consisting of 152 layers. Its architecture led to significant improvements in classification, detection and localization techniques. ResNet architecture consists of a set of subsequent residual modules which are the basic building blocks. End to end network is formed by stacking the residual networks on top of each other. The technique of input preprocessing is changed in ResNet. Input is first divided into patches before feeding into the network. It overcomes two significant shortcomings of previous models – vanishing gradient and degradation. ResNet is considered an effective because during backward pass of propagation the gradient flows easily [8]. In simple words, a Reset has two options, it can either perform a set of functions on the input or it can altogether skip the step. It has an error rate of 3.6%. It consists of mostly 3×3 filters like VGG and uses global average pooling followed by classification. The main advantage of ResNet is that even thousands of residual layers can be used to create a network and then trained.

3 Transfer Learning

The amount of data required for training a convolutional neural network from scratch is very high but sometimes we cannot obtain a large dataset. To overcome this disadvantage the concept of transfer learning is used. In transfer learning a pre-trained model that has been trained using a large dataset before is used for training the network with a comparatively small dataset. This pre-trained model acts as a feature extractor for the new dataset. The weights in the previous layers are freezed, the last layer is replaced by a new classifier and the network is trained normally. An example of large dataset is Imagenet [9] which contains about 1.2 million images with 1000 classes.

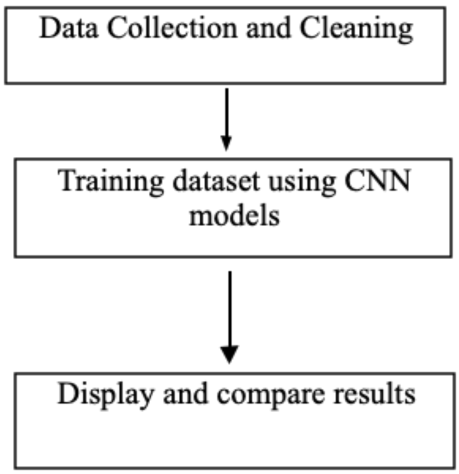

Fig. 2 Representation of basic workflow

Since the lower layers of this network detect more common features but in higher layers the features become more specific to the dataset therefore the weights of the pre-trained network of all or some layers can also be fine- tuned by continuing the process of back propagation. Transfer learning should be applied taking into consideration the size of the new dataset along with the similarity to the original dataset.

4 Literature Survey

Sign language recognition has been implemented by researchers using numerous algorithms. In [10] two Markov models were presented to recognize American Sign language (ASL) with the help of a single camera for tracking users hand. The first system achieved around 92% accuracy and the accuracy of the second system was about 98%. [11] shows promising results for recognizing asl through parallel hidden markov models(HMM) and demonstrates that even on a small scale it can improve hmm based models' robustness. [12] uses skin colors for extracting face and hand regions for tracking the location of each hand of the person performing hand gestures. Their experimental results show that hands can be tracked even if they are overlapping the face. In [13] signs were broken down into their building blocks or phonemes using movement-hold model. Hmm and parallel Hmm were used to conduct experiments on 22−sign set. Microsoft Kinect based system was compared in [14] to their own system (copycat) which consisted of accelerometers and colored gloves for tracking hands. Their results showed that Kinect is a better option for SLR. [15] also used Kinect for Chinese sign language (CSL) recognition and translation and obtained promising results. Transition models were introduced in [16] to handle transition between adjacent signs in continuous SLR. A temporal clustering algorithm was also proposed which is an improvement over k-means algorithm for dynamically clustering signs. They obtained an accuracy of 91.9% on CSL. In [17] Gabor filters and support vector machine (svm) were used for recognizing hand gestures and an accuracy of 95.2% was obtained. A system was developed in [18] for translating Arabic sign language using neuro fuzzy algorithm which obtained an accuracy of 93.55%. A Bosnian sign language translator was developed in [19] which used digital image processing methods for feature extraction and multilayer neural network for training thus achieving an accuracy of 84%. In [20] a gesture recognition system was developed using Recurrent Neural network for handling dynamic gestures and encouraging results were obtained. American Sign Language finger spelling translator based on pre-trained googleNet architecture was implemented in [21]. A robust model that correctly classified letters $a - e$ was identified and a system that correctly recognized letters $a - k$ except letter j most of the times was also produced. [22] evaluated the different convolutional neural networks

Table 1 Dataset Description

Number of Images	1791
Number of Rows	403
Number of Columns	298

on the Marcel dataset. The best accuracy was obtained using the GoogleNet architecture followed by their custom proprietary model which was designed for pixel-based segmentation of images and the obtained accuracy was 64.17%. A real time hand gesture recognition system was developed for Indian Sign Language recognition. The system comprises four modules: real-time hand tracking, hand segmentation, feature extraction, and gesture recognition, which is implemented using a genetic algorithm. Further, deep learning, image segmentation, clustering, character recognition, also considered for insight in literature [23, 24].

5 Experimental Setup

The performance measure of various CNN models namely GoogleNet, Resnet and VGG has been compared on Massey University Gesture dataset [25] using TensorFlow and Python. The dataset consists of 1791 images from 5 users of hand signs. These colored images are of an alphabet $(a - z)$ and are cropped such that the hands touch all four edges of the frame. Every Massey data example consists of an image as well as its correct label $(a - z)$. The images are of size 298×403 pixels as shown in Table 1. (The value of each pixel is between $0 - 255$.) Transfer learning is used to train these models. The last layer of each model is replaced by a classifier formed through our dataset and then the whole model is trained. We employ the standard Softmax function in the last layer of our classifier such that for a test image it predicts probabilities for each label as output. We train these models for 500 epochs and measure these models on the basis of how accurately they recognize a gesture and label it correctly. The various steps involved in training our dataset on different models are given in Figure 2.

5.1 Data Collection and Cleaning

Data was collected from Massey University's website which consists of Images of 5 users of hand gestures. We divided these images according to 26 labels $(A - Z)$.

5.2 Training dataset using CNN models

Training was done using Docker and Tensorflow. During the first phase, bottleneck values for each image were calculated. After all the files were created, the actual training of the final layer began. The training operates efficiently and takes less time since we are feeding the cached value for each image into the bottleneck layer.

6 Results

According to our experimental results GoogleNet performed better than Resnet and VGG as shown in Table 2 and Figure 3 because it is not completely linear (layers stacked on top of each other) but works parallely. Since we don't know the size of a convolution (for example 3×3 or 5×5) that will work better for our model, GoogleNet combines all these convolution to work parallely and concatenates the results that act as an input to the next layer. In this way the model has options to choose from and decides what's best for it. Also this architecture acts like a multi feature extractor, extracting local features from smaller convolutions and high level features from larger convolutions.

Table 2 Comparison of Different CNN Models

Model	Accuracy (%)
GoogleNet	90.2
ResNet	89.3
VGG	89.17

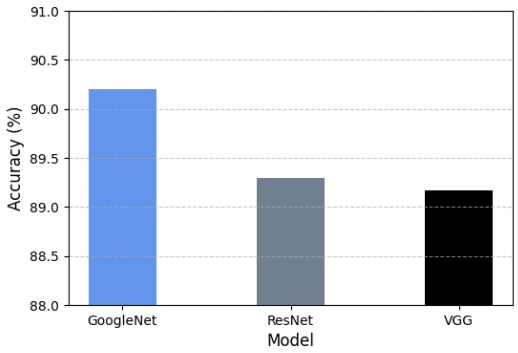

Fig. 3 Comparison of Different CNN Models in terms of Percentage Accuracy

7 Conclusion and Future Scope

In terms of training accuracy, GoogleNet performed best because it uses global average pooling instead of fully connected layers. This helps in averaging out the channel values and also leads to a significant decrease in the number of parameters. Although we are able to achieve phenomenal accuracies with these models we conclude that huge computational requirements both in terms of memory and time are required for achieving best results which are possible with system resources like GPUs. In future the system can be extended to work in both directions i.e. from sign language to normal language and vice versa. We will also recognize signs that involve motion. The system can further be made portable so that it can help in communication on the go. It can be implemented in real time so that the images are captured through webcam and the output of sign language will be displayed in text form in real time.

Declarations

- The authors received no specific funding for this study.
- The authors declare that they have no conflicts of interest to report regarding the present study.
- No Human subject or animals are involved in the research.
- All authors have mutually consented to participate.
- All the authors have consented the Journal to publish this paper.
- Authors declare that all the data being used in the design and production cum layout of the manuscript is declared in the manuscript.

References

[1] Mishra, G., Vishwakarma, V.P.: Nested sparse classification method for hierarchical information extraction. In: Proceedings of International Conference on Artificial Intelligence and Applications: ICAIA 2020, pp. 533–542 (2021). Springer

[2] Mittal, A., Kumar, D.: AiCNNs (Artificially-integrated Convolutional Neural Networks) for Brain Tumor Prediction. EAI Endorsed Transactions on Pervasive Health and Technology

5(17) (2019)

[3] Kumar, D., Batra, U.: Clustering algorithms for gene expression data: A review. International Journal of Recent Research Aspects **4**, 122–128 (2017)

[4] Boureau, Y.-L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010)

[5] CS231n: Convolutional Networks. http://cs231n.github.io/convolutional-networks/

[6] Mishra, G., Vishwakarma, V.P.: A robust two quadrant sparse classifier for partially occluded face image recognition. Journal of Discrete Mathematical Sciences and Cryptography **23**(5), 1047–1057 (2020)

[7] He, K., *et al.*: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)

[8] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009). Ieee

[9] Starner, T., Weaver, J., Pentland, A.: Real-time american sign language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(12), 1371–1375 (1998)

[10] Vogler, C., Metaxas, D.: Parallel hidden markov models for american sign language recognition. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 1, pp. 116–122 (1999). IEEE

[11] Imagawa, K., Lu, S., Igi, S.: Color-based hands tracking system for sign language recognition. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 462–467 (1998). IEEE

[12] Vogler, C., Metaxas, D.: Handshapes and movements: Multiple-channel american sign language recognition. In: Gesture Workshop vol. 2915, (2003)

[13] Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P.: American sign language recognition with the kinect. In: Proceedings of the 13th International Conference on Multimodal Interfaces, pp. 279–286 (2011)

[14] Yadav, M., Purwar, R.K., Mittal, M.: Handwritten hindi character recognition-a review. IET Image Processing **12**(11), 1919–1933 (2018)

[15] Fang, G., Gao, W., Zhao, D.: Large-vocabulary continuous sign language recognition based on transition-movement models. IEEE transactions on systems, man, and cybernetics-part a: systems and humans **37**(1), 1–9 (2007)

[16] Huang, D.-Y., Hu, W.-C., Chang, S.-H.: Vision-based hand gesture recognition using pca+ gabor filters and svm. In: 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 1–4 (2009). IEEE

[17] Al-Jarrah, O., Halawani, A.: Recognition of gestures in arabic sign language using neuro-fuzzy systems. Artificial Intelligence **133**(1–2), 117–138 (2001)

[18] ogić, S., Karli, G.: Sign language recognition using neural networks. TEM Journal **3**(4), 296–301 (2014)

[19] Murakami, K., Taguchi, H.: Gesture recognition using recurrent neural networks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 237–242 (1991)

[20] Garcia, B., Viesca, S.A.: Real-time American Sign Language Recognition with Convolutional Neural Networks (2019)

[21] Strezoski, G., et al.: Hand Gesture Recognition using Deep Convolutional Neural Networks (2020)

[22] Ghotkar, A.S., Khatal, R., Khupase, S., Asati, S., Hadap, M.: Hand gesture recognition for indian sign language. In: 2012 International Conference on Computer Communication and Informatics, pp. 1–4 (2012). IEEE

[23] Aggarwal, A., Chauhan, A., Kumar, D., Mittal, M., Roy, S., Kim, T.H.: Video caption based searching using end-to-end dense captioning and sentence embeddings. Symmetry **12**(6), 992 (2020)

[24] Goyal, M., Malik, R., Kumar, D., Rathore, S., Arora, R.: Musculoskeletal abnormality detection in medical imaging using gncnnr (group normalized convolutional neural networks with regularization). SN Computer Science **1**(6), 1–12 (2020)

[25] University, M.: Gesture Dataset. http://www.massey.ac.nz/~albarcza/gesture_dataset2012.html