Next Generation Computing Technologies 2019

Sign Language Recognition Using ResNet50 Deep Neural Network Architecture

Pulkit Rathi[a], Raj Kuwar Gupta[a], Soumya Agarwal[a], Anupam Shukla[b], Ritu Tiwari[b]

[a]ABV – IIITM, Gwalior, Madhya Pradesh, India - 474010

[b]IIIT, Pune, Maharashtra, India - 411048

ARTICLE INFO

Article history:
Received 02 October 19
Received in revised form 02 October 19
Accepted 15 November 19

Keywords:
convolution neural network,
gesture classification,
resnet50,
sign language recognition,
transfer learning.

ABSTRACT

Communication is a barrier between the deaf-mute community and the rest of the society. Sign Language is used for communication among such people who cannot speak and listen. The automation of sign language recognition has gained researchers attention in the last few years. Many complex and costly hardware systems have already been developed to assist the purpose. However, we propose to use deep learning approach for automated sign language recognition. We devised a novel 2-level ResNet50 based Deep Neural Network Architecture to classify fingerspelled words. The dataset used is the standard American Sign Language Hand gesture dataset by (Barczak et al., 2011). The dataset was first augmented using various augmentation techniques. In our 2-level ResNet50 based approach the Level 1 model classifies the input image into one of the 4 sets. After an image is classified into one of the sets it is provided as an input to the corresponding second level model for predicting the actual class of the image. Our approach yields an accuracy of 99.03% on 12,048 test images.

Peer review under responsibility of UPES Dehradun and NGCT 2019.

Introduction

Sign language is the language of the deaf and mute. However, this population of the world is unfortunately overlooked because of the lack of communication as sign language is not understood by the majority hearing population. Thus there exists a communication gap and it is difficult to incorporate them into the mainstream society due to which they feel sidelined. Sign language has three major classes:

Fingerspelling: The signer makes gestures for each letter used to spell the word.

Word Level Sign Vocabulary: Hand gestures for words are used for communication.

Non-manual features: Here the signer uses a combination of facial expressions, tongue, mouth and body poses to communicate.

The main focus of this work is to create a vision based system to identify fingerspelled letters of sign language. The field of computer vision is focused on simplifying and generalizing the way of communication between a human and a computer. Advances in this field have led to the birth of self-driving cars and other futuristic technologies. We aim to leverage the advances in the area of computer vision to bridge the communication gap so that the deaf and mute people can exchange their ideas with everyone and don't feel neglected.

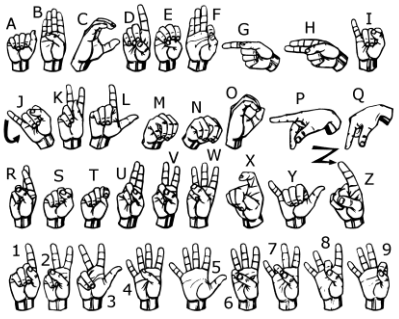

Fig. 1 - ASL fingerspelling alphabets and numbers (Anon, 2019).

1. Previous Work

A lot of work has been done in the field which includes sensor-based methods for sign detection. These include Kinect sensor (Yang et al., 2014) and glove based methods (Mehdi et al., 2002). In glove based methods, the hand gesture detection is very accurate. The sensors directly track the gestures made by the fingers using finger position and joint mapping and thus their accuracy of hand pose detection is high. Kinect sensor is a special purpose camera which captures the depth in the images and helps in gesture detection with high accuracy. However, the arrangement of both these systems is complex and have limited applications in the practical scenario.

In (Raheja et al., 2016) the authors employed SVM for recognition of four characters ('A', 'B', 'C', 'Hello') in Indian Sign Language. They converted the video frames to HSV (Hue Saturation Value) colour space and then segmented them on the basis of skin pixels. Kinect sensor is used to capture the image. They extracted hue moments from the image frames and classified the images using SVM.

In (Misra et al., 2011), the authors proceeded by extracting Histogram of Oriented Gradients from the images. Thereafter they used partial least square regression for dimensionality reduction and projected the image in 10 dimensions. They employed KNN to compare this reduced feature map with those present in their database to predict the class label.

In (Dong et al., 2015) the authors used a color glove and Kinect sensor to obtain the depth image of a hand. The color glove helped in obtaining an 11 part segmented hand: 5 fingertips, 5 lower section of fingers and palm after which a per-pixel classification was done using a Random Forest classifier. The colour glove also helped in identifying the joint angle features and mapping the interrelationships for a particular class label.

Deep Learning gained traction after the unprecedented success of AlexNet (Krizhevsky et al., 2012) at ILSVRC-2012 where the 8-layer CNN beat its competitors by a large margin. In the following years, researchers experimented with various architectures and the general trend was to increase the depth and width of the network for higher representation power. (Simonyan et al., 2014) gave a very simple yet powerful 16 layer CNN architecture, VGG-16, having significant accuracy gains on ImageNet (Deng et al., 2009) dataset but almost double the parameters than AlexNet. (Szegedy et al., 2015) gave GoogLeNet, a 22 layer CNN architecture that captures features at different scales simultaneously to boost the representation power. GoogLeNet employed 1 X 1 convolution operations for dimensionality reduction and had much lower parameters than VGG16 and maintained a reasonable computation budget of 1.5 Billion Flops. (Ioffe et al., 2015) proposed batch normalization technique to tackle the problem of internal covariate shift which tends to increase the time required to train a network as lower learning rates are required to be used. Batch normalization makes it possible to train much deeper networks on existing hardware. He et al. in (He et al., 2016) introduced "skip connections" in their ResNet architecture which addressed the problem of deteriorating representation power of deeper networks

In (Pan et al., 2010), the authors studied the relationship between transfer learning and other machine learning approaches like domain adaptation, multitask learning and sample selection bias. (Yosinski et al., 2014) found out that the transferability of features decreases as the dissimilarity between the base task and target task increases. However, transfer learning is always a better approach than random initialization of the learnable parameters.

2. Methodology

We divide our work into two parts:

Data Division and Augmentation: Complete dataset was first divided into three sets, Train, Test and Validate. Further augmentation techniques were used to increase the size of the dataset.

Classification: A deep convolutional neural network based on the original ResNet50 architecture is used to predict the class of the image.

2.1. Dataset

The American Sign Language dataset is taken from the Gesture Dataset 2012 from Massey University, Institute of Information and Mathematical Sciences (Barczak et al., 2011).

The dataset originally contains 2515 images of the 26 alphabets and 10 digits from 5 different users and in varying light effects.

2.2. Dataset Augmentation

We first split the dataset in the ratio of 70:10:20 for training, validation and test data so that image augmentation does not introduce correlation in the training and test images. As already stated, originally we had 2515 RGB images with approx. 70 images in each of the 36 categories. All the images in the dataset were of varying sizes and so the first step in data augmentation was resizing of the images to 224 X 224 pixels maintaining the original aspect ratio of the image.

Fig. 2 - Scaled images for alphabet 'A'

Fig. 3 – Flipping over Scaled images

Fig. 4 – Adding Salt and Pepper Noise

Fig. 5 – Changed Lighting effect

After resizing, the images were scaled in three different sizes to address the problem of variable distance of camera module from signer. The images were then flipped to incorporate detection of signs made by left-handed signers. Random salt and pepper noise was added to incorporate the effect of instability of camera while taking the images as well as to reduce dependency on any single pixel. Thereafter Random Gaussian Noise was added to the images generated so far to simulate different lighting conditions. After completion, we have 24 images for each image in the original dataset. Image augmentation was a necessary step as it helped in increasing the size of the dataset by producing several images from a single image to handle the problem of over-fitting. This will help the classifier to perform on the images when different camera modules are used or lighting conditions are different. Data augmentation also helped us to introduce more robustness to the classifier.

2.3. ResNet50

After the success of AlexNet, VGG16, InceptionV3, it may seem that deeper neural networks perform better. However, if we keep on stacking convolution layers in a linear fashion, we will find that not only the time and memory requirements of the network tend to increase but also the networks perform worse. This is because the problem of finding the optimal weights becomes increasingly difficult as the depth of the network increases.

The ResNet architecture introduced in (Pan et al., 2010), tackles this degradation problem and achieve much more depth. The Residual networks are deep neural networks that follow the basic idea of skipping blocks using the shortcut connections. The blocks follow two simple design structures:

- For a same output feature map size, the number of filters remains same.
- If the size of the feature map is halved, the number of filters is doubled.

In case of same input and output dimensions, identity shortcut is used otherwise projection shortcuts are used. The basic building block of ResNet architecture is summarized by the following equation:

$$y = F(x, \{Wi\}) + x \qquad (1)$$

where x and y are input and output vectors of the convolution layer under consideration. The function F(x, {Wi}) represents the residual mapping learned. The dimensions of x and F should be equal in equation (1).

Fig. 6 – Transfer Learning

2.4. Initial Approach

For the first level model, the dataset was divided into four sets of nine classes each. No particular approach was followed in dividing the 36 classes into these 4 sets. The 4 sets contains classes as follows:

1. Set 1: Classes 0-8
2. Set 2: Classes 9 and A-H
3. Set 3: Classes I-Q
4. Set 4: Classes R-Z

The Level 1 model classifies the input image into one of these sets. After an image is classified into one of the sets it is provided as an input to the corresponding second level model for predicting the actual class of the image.

We fine-tuned ResNet50 model pre-trained on the ImageNet dataset. We tried several configurations by adding and dropping layers and the best performing configuration on the validation set is described below:

For the first level model, we dropped last 3 residual blocks of the original ResNet50 model. After these 4 more layers were added to adapt our model to classify hand gestures. These layers include a convolution layer of 1 X 1 with 512 filters and stride of two, a flatten layer, a dense layer with 128 neurons followed by a dropout with probability 0.3 and a final dense layer which outputs the final prediction. The network was trained using 128 images per batch. Each image is normalized before giving it as input to the model. The output of this model is used to choose the second level model which will finally predict the actual class of the image. It is important to note here that if the first level model makes a wrong prediction then the final prediction made by the second level model is bound to be incorrect hence the first level model acts as the bottleneck for our classification method. Now we train second level models separately using the same ResNet50 architecture as described above. The only variation is the slight increase in the dropout probability to 0.50 and 32 images per batch were used while training this model which we found to be best performing on the validation set.

2.5. Improvised Approach

After analysing the confusion matrix of our initial results, we tried to improve the bottleneck accuracy of our Level 1 model by trying various combinations of the classes after which we arrived at the best performing combination on the validation set which is as follows:

1. Set 1: Classes Z, 1, 4, 5, 6, 7, 8, C, W
2. Set 2: Classes 3, 9, A, B, D, R, F, G, H
3. Set 3: Classes I, J, K, O, 0, X, Y, P, Q

4. Set 4: Classes E, S, T, M, N, U, V, 2, L

The rest of the approach is the same as described in Section 2.4. The reasons for this grouping is discussed in Section 3.

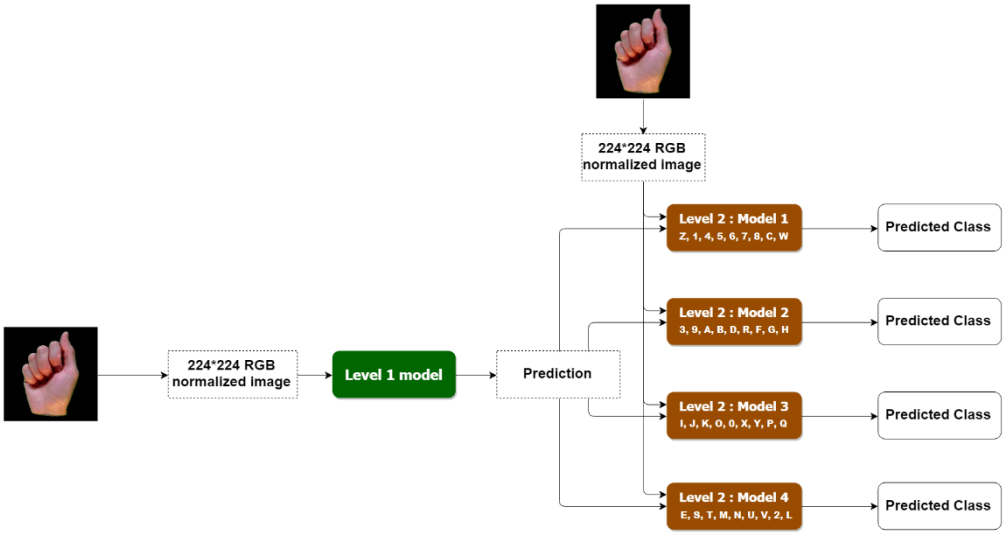

Fig. 7 – Improvised Approach

3. Experimental result

We evaluate the performance of the proposed ResNet50 model on the described dataset. For classification purpose, we first train the above described model on the training set. The training is done using 'categorical cross entropy' loss function and 'SGD' optimizer with a learning rate of 1e-4, decay 1e-6 and momentum 0.9 on Nvidia GeForce GTX 1080 12GB GPU. After the complete training of the first level and second level models, these models are integrated to find out the combined final accuracy.

The second level models are trained independently of the first level models. We reiterate that the prediction from first level model is solely used to choose the second level model to make the final prediction and no output is passed from the first level model to second level models for making predictions.

Table 1 – Comparison table.

Technique	Accuracy
2-level ResNet50 (Proposed approach)	99.03%
VGG16 (Masood et al., 2018)	95.20%
Sparse Autoencoders (Kala et al., 2014)	83.6%
Gesture Segmentation (Kulkarni et al., 2010)	92.33%
deepCNN (Bheda et al., 2017)	82.5%

3.1. Evaluation of Performance

3.1.1 Initial Approach

We achieve test accuracy of 98.48% on the test dataset of 12,048 test images i.e. 11,865 images were correctly classified. The individual accuracy of the Level 1 model is presented in Table 3.2 and the respective accuracies of Level 2 models are presented in Table 3.3. We further investigate the performance of our methodology by plotting a confusion matrix as shown below. It can be clearly seen from the confusion matrix that most of the

characters have been classified correctly with a few exceptions which we explain below:

1) Class ‘0’ and Class ‘O’
2) Class ‘2’ and Class ‘V’
3) Class ‘6’ and Class ‘W’

There is a very high degree of similarity between the aforementioned pair of classes. Class ‘0’ and Class ‘O’ have very subtle differences. Class ‘2’ and Class ‘V’ differ only in the position of the thumb. In ‘2’ the thumb resides on the second knuckle of ring finger while in ‘V’ it resides on the nail of the ring finger. Class ‘6’ and Class ‘W’ also differ only in the position of the thumb.

Table 2 – Accuracy of Level 1 model (Initial Approach)

ACCURACY	TRAINING TIME (MIN)	EPOCHS
98.58	320	200

Table 3 – Accuracy of various Level 2 models (Initial Approach)

MODEL	ACCURACY	TRAINING TIME (MIN)	EPOCHS
Classes 0-8	99.60	715	20
Classes 9, A-H	99.44	723	20
Classes I-Q	99.48	732	20
Classes R-Z	99.37	719	20

3.1.2 Improvised Approach

Since Level 1 model serves as the bottleneck it seems reasonable to put similar images in the same group so that we can overcome the bottleneck created by the Level 1 model. By following this approach, we achieve test accuracy of 99.03% on the test dataset of 12,048 test images i.e. 11,931 images were correctly classified. The individual accuracies of the Level 1 model is presented in Table 4 and the respective accuracies of level 2 models are presented in Table 5. We further investigate the performance of our methodology by plotting a confusion matrix as shown below.

Table 4 – Accuracy of Level 1 model (Improvised Approach)

ACCURACY	TRAINING TIME (MIN)	EPOCHS
99.79	316	150

Table 5 – Accuracy of various Level 2 models (Improvised Approach)

MODEL	ACCURACY	TRAINING TIME (MIN)	EPOCHS
Classes Z, 1, 4, 5, 6, 7, 8, C, W	99.10	711	20
Classes 3, 9, A, B, D, R, F, G, H	99.80	725	20
Classes I, J, K, O, 0, X, Y, P, Q	99.70	720	20
Classes E, S, T, M, N, U, V, 2, L	98.07	716	20

4. Conclusion and Future Scope

Sign language recognition is a field of main focus for many researchers. Several sensor based systems already exist. However, these devices are quite expensive and their deployment seems impractical. We have developed a computer vision based approach for recognition of sign language. Our work presents a novel 2-level Res-Net50 based Neural Network Architecture for Fingerspelling based Sign Language Recognition using the image dataset for American Sign Language. Our work shows that the software- based techniques can also be used for effective classification. The dataset containing 60,336 image samples from 36 classes of 10 digits and 26 alphabets was used for evaluation and we obtained an accuracy of 99.03% on test set. Also, the average time for prediction is about 25ms and does not require any preprocessing in a bounding box based implementation which makes this ideal for practical applications. The approach presents a way of recognizing the character from a given image. The image is given as an input and our classifier successfully categorizes it into the character it represents. These alphabets when made one after the other can be used to spell complete words. Effective extension of the proposed methodology to words and common expressions can make the deaf and mute people communicate in a fast and efficient manner with the rest of the society. This approach can be ex-tended for recognition of other Sign Language variants including the Indian Sign Language by collecting a similarly sized dataset. Also, this technique can be extended to develop gesture-controlled smart devices to provide more intuitive interfaces for mainstream adoption.

REFERENCES

Barczak, A. L. C., Reyes, N. H., Abastillas, M., Piccio, A. and Susnjak, T.: 2011, A new 2d static hand gesture colour image dataset for asl gestures., Res Lett Inf Math Sci .

Anon, (2019). [online] Available at: http://www. lifeprint.com/asl101/fingerspelling [Accessed 5 Dec. 2019].

Yang, H.D., 2014. Sign language recognition with the kinect sensor based on conditional random fields. *Sensors*, *15*(1), pp.135-147.

Mehdi, S.A. and Khan, Y.N., 2002, November. Sign language recognition using sensor gloves. In *Neural Information Processing, 2002. ICONIP'02. Proceedings of the 9th International Conference on* (Vol. 5, pp. 2204-2206). IEEE.

Raheja, J.L., Mishra, A. and Chaudhary, A., 2016. Indian sign language recognition using SVM. *Pattern Recognition and Image Analysis*, *26*(2), pp.434-441.

Misra, A., Abe, T. and Deguchi, K.: 2011, Hand gesture recognition using histogram of oriented gradients and partial least squares regression., MVA, pp. 479– 482.

Dong, C., Leu, M. C. and Yin, Z.: 2015, American sign language alphabet recognition using Microsoft Kinect, Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 44–52.

Krizhevsky,A.,Sutskever,I.andHinton,G.E.: 2012,Imagenetclassificationwith deep convolutional neural networks, Advances in neural information processing systems, pp. 1097–1105.

Simonyan, K. and Zisserman, A.: 2014, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 .

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

Szegedy,C.,Liu,W.,Jia,Y.,Sermanet,P.,Reed,S.,Anguelov,D.,Erhan,D.,Vanhoucke, V. and Rabinovich, A.: 2015, Going deeper with convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.

Ioffe, S. and Szegedy, C.: 2015, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167 .

He, K., Zhang, X., Ren, S. and Sun, J.: 2016, Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Pan, S.J., Yang, Q. et al.: 2010, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22(10), 1345–1359.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H.: 2014, How transferable are features in deep neural networks? , Advances in neural information processing systems, pp. 3320–3328.

Masood, S., Thuwal, H. C. and Srivastava, A.: 2018, American sign language character recognition using convolution neural network, Smart Computing and Informatics, Springer, pp. 403–412.

Kala, R., Nandi, G. C. and Kumar, V.: 2014, Static hand gesture recognition using stacked denoising sparse autoencoders, Contemporary Computing (IC3), 2014 Seventh International Conference on, IEEE, pp. 99–104.

Kulkarni, V. S. and Lokhande, S.: 2010, Appearance based recognition of American signlanguage using gesture segmentation, International Journal on Computer Science and Engineering 2(03), 560–565.

Bheda, V. and Radpour, D.: 2017, Using deep convolutional networks for gesture recognition in american sign language, arXiv preprint arXiv:1710.06836 .