Detection of Sign Language Through Gesture Recognition

Ravi Maithrey Regulagedda[1], Ippili Akarsh[2], Chinta Aaron John[3], Dr.K Manikandan[4*]
[1−4] School of Computer Science and Engineering, Vellore Institute of Technology,
Vellore - 632014, Tamil Nadu
[*]*kmanikandan@vit.ac.in*

***Abstract*—Deaf people use sign languages to communicate with other people in the community. Although the sign language is known to hearing-impaired people due to its widespread use among them, it is not known much by other people. In this paper, we have developed a real-time sign language recognition system for people who do not know sign language to communicate easily with hearing-impaired people. The sign language used in this paper is American sign language. In this paper a neural network based on the MobileNet Architecture was trained to try for a positive performance with the advantages given by that MobileNet.**

***Index Terms*—Sign Language Detection, Deep Learning, Classification, Convolutional Neural Networks, MobileNet**

I. Introduction

Communication is of great importance to us as a species, and personally, we do it by talking and listening to people talk. But for the deaf and mute among us, the only way they are able to communicate is through sign language This sign language is in the form of hand gesture, with each gesture referring to either a letter, a word, or an action. This forms a great barrier between people who can and can't understand the sign language being used. So, there is a need for the sign language to be translated Sign language recognition is a problem that has been addressed in research for years. However, we are still far from finding a complete solution available in our society.

Among the works developed to address this problem, the majority of them have been based on basically two approaches: contact-based systems, such as sensor gloves; or vision-based systems, using only cameras. The latter is way cheaper and the boom of deep learning makes it more appealing.

Vision is a key factor in sign language, and every sign language is intended to be understood by one person located in front of the other, from this perspective, a gesture can be completely observable. Viewing a gesture from another perspective makes it difficult or almost impossible to be understood since every finger position and movement will not be observable.

For this reason, any possible solution to the problem of sign language recognition needs to take into consideration the appearance of different signs and how they can be represented in a static image. It is this static image which can be classified and shown into different classes of the alphabets it represents.

II. Prior Work

This paper [1] proposes a new methodology to map images in a medical scenario. They propose a methodology based on efficient Convolution Neural Network (CNN) architecture in order to classify epidemic pathogen with five deep learning phases: (1) Training dataset of provided images (2) CNN Training (3) Testing data preparation (4) CNN generated model on testing data and finally (5) Evaluation of images classified.

Although this document addresses the classification of epidemic pathogen images using a CNN model, the underlying principles apply to the other fields of science and technology, because of its performance and its capability to handle more layers than the previous traditional neural networks.

Three major techniques that successfully employ CNNs to medical image classification are from this paper by Simonyan et al. [2]. The authors talk about training the CNN from scratch, using off-the-shelf pre-trained CNN features, and conducting unsupervised CNN pre-training with supervised fine-tuning. Another effective method is transfer learning, i.e., fine-tuning CNN models pre-trained from natural image dataset to medical image tasks. In this paper, we can see how the authors exploit three important, but previously understudied factors of employing deep convolutional neural networks to computer-aided detection problems.

The authors first explore and evaluate different CNN architectures. The studied models contain 5 thousand to 160 million parameters, and vary in numbers of layers. Then they evaluate the influence of dataset scale and spatial image context on performance. Finally, they examine when and why transfer learning from pre-trained ImageNet (via fine-tuning) can be useful.

Another paper [3] offers a state-of-the-art look at sign language recognition. Computer recognition of sign language deals from sign gesture acquisition and continues till text/speech generation. Sign gestures can be classified as static and dynamic. However static gesture recognition is simpler than dynamic gesture recognition but both recognition systems are important to the human community. The sign language recognition steps are described in this survey. The data acquisition, data preprocessing and transformation, feature extraction, classification and results obtained are examined. Some future directions for research in this area also suggested.

In this work [4] the authors investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Their main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16-19 weight layers.

The authors also show that these representations generalise well to other datasets, where they achieve state-of-the-art results.

This seminal paper [5] presents MobileNets. The authors present a class of efficient models for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. They introduce two simple global hyper-parameters that efficiently trade off between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem.

They also present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification. Then the authors demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, fine-grain classification, face attributes and large scale geo-localization.

The next paper [6] considers a recognition system using the Microsoft Kinect, convolutional neural networks (CNNs) and GPU acceleration. Instead of constructing complex handcrafted features, CNNs are able to automate the process of feature construction. The authors were able to recognize Italian gestures with high accuracy. The predictive model is able to generalize on users and surroundings that do not occur during training with a cross-validation accuracy of 91.7%. This paper presents a sample of how a basic method for hand gesture recognition might work.

The next important work is YOLO [7]. Prior work on object detection depends on classifiers to be trained to perform the detection. Instead, this paper frames object detection as a regression problem to create space-separated bounding boxes with probabilities for each based on the possible object detected. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance.

Compared to state-of-the-art detection systems, YOLO makes more localization errors but is far less likely to predict false detections. Finally, YOLO learns very general representations of objects. It outperforms all other detection methods, including DPM and R-CNN, by a wide margin when generalizing.

Deep Neural Networks (DNNs) have recently shown outstanding performance on image classification tasks. In this paper [8] the authors go one step further and address the problem of object detection using DNNs, that is not only classifying but also precisely localizing objects of various classes. They present a simple and yet powerful formulation of object detection as a regression problem to object bounding box masks. They define a multi-scale inference procedure which is able to produce high-resolution object detections at a low cost by a few network applications
In this paper [9], the authors provide a review on deep learning based object detection frameworks. This review begins with a brief introduction on the history of deep learning and its representative tool, namely Convolutional Neural Network(CNN). Then they focus on typical generic object detection architectures along with some modifications and useful tricks to improve detection performance further.

As distinct specific detection tasks exhibit different characteristics, the authors also briefly survey several specific tasks, including salient object detection,face detection and pedestrian detection. Experimental analyses are also provided to compare various methods and draw some meaningful conclusions. Finally, several promising directions and tasks are provided to serve as guidelines for future work in both object detection and relevant neural network based learning systems

III. Proposed Work

A. Objectives

In this paper, we aim to come up with a model for hand gesture recognition using a MobileNet based architecture. This architecture was chosen due it's ease of use in mobile devices and embedded architectures. This will allow it to be deployed to any number of mobile devices and devices with low processing power.

The aim is to be able to develop a model which, once trained, will be able to be deployed in systems in remote places to aid in the ease of communication.

B. System Design

The user must be able to capture images of the hand gesture using web camera and the system shall predict and display the name of the captured image. We use the HSV colour algorithm to detect the hand gesture and set the background to black. The images undergo a series of processing steps which include various Computer vision techniques such as the conversion to grayscale, dilation and mask operation.

And the region of interest which, in our case is the hand gesture is segmented. The features extracted are the binary pixels of the images. We make use of Convolutional Neural Network (CNN) for training and to classify the images.

Identification of sign gesture is performed with either of the two methods. First is a glove-based method whereby the signer wears a pair of data gloves during the capture of hand movements. In our case, the glove based method was discarded due the reasons described above. Since the objective was to develop a model which works anywhere, having a glove to go with that would be prohibitive.

And despite having an accuracy of over 90%, wearing of gloves are uncomfortable and cannot be utilised in rainy weather. They are not easily carried around since their use require computer as well. Therefore we opted for the second method

Second is a vision-based method, further classified into static and dynamic recognition. Static deals with the 2D representation of gestures while dynamic is a real time live capture of the gestures. In this case, we have decided to go with the static recognition of hand gestures because it increases accuracy as compared to when including dynamic hand gestures

C. Architecture of the proposed model

Mobilenet is a neural network architecture which is primarily used for feature extraction and for supporting classification. The idea behind mobilenet is to develop a neural network architecture which is both deep enough to allow for significant learning but also lightweight enough that it can work on any small device. Mobilenet was chosen in order to create this model which works on all sorts of devices to ensure this hand gesture recogniser is accessible to all.

Depthwise Separable Convolution is used to reduce the model size and complexity. It is particularly useful for mobile and embedded vision applications because -

- **Smaller model size:** Fewer number of parameters
- **Smaller complexity:** Fewer Multiplications and Additions (Multi-Adds)

The main ideas in MobileNet Architecture are

1) Depthwise Separable Convolution: Depthwise separable convolution is a depthwise convolution followed by a pointwise convolution as follows:

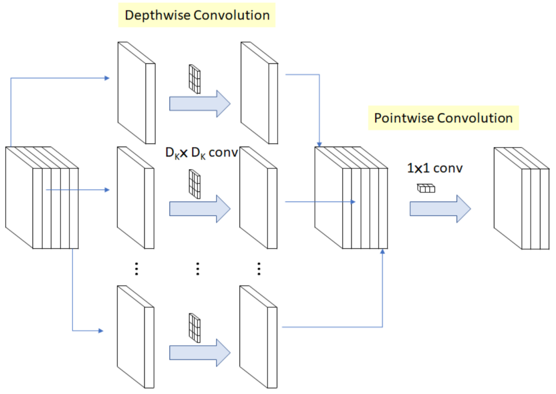
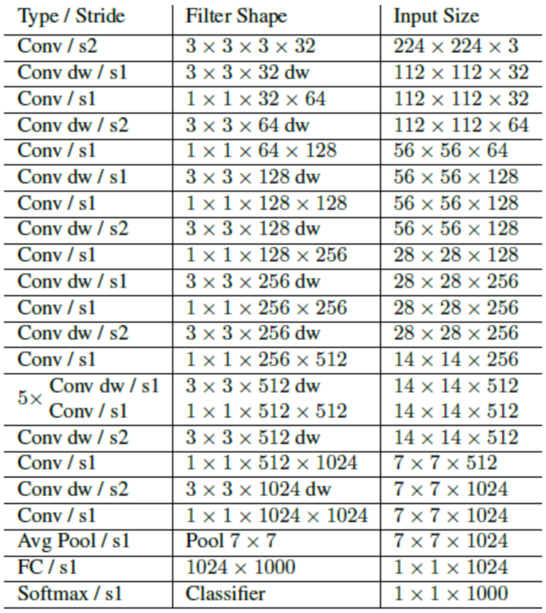


Fig. 1. Depthwise Separable Convolution

- Depthwise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, we have 5 channels, then we will have 5 DK×DK spatial convolution.
- Pointwise convolution actually is the 1×1 convolution to change the dimension.

When DK×DK is 3×3, 8 to 9 times less computation can be achieved, but with only small reduction in accuracy.

2) MobileNet Architecture: The architecture of the MobileNet is described in the table below



Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig. 2. Different layers in the MobileNet

In this architecture, batch normalization and ReLU are applied after every convolution layer. Adding this to the Depthwise Seperable Convolution layers gives us a huge reduction in the number of multiplication and additions steps to be performed in each layer and its calculation.

The final few layers are first an average pool layer for bringing together the calculations, then a compression layer and finally the actual classification itself. The classification is done on the calculated values by means of a softmax layer. This softmax layer outputs the class number which has been calculated with the highest probability.

In our case, it takes in as a vectors, the calculated probabilities and normalizes it into a probability distribution over the total classes present. The class which has the highest value in this probability distribution is the predicted class.

D. Dataset Used

This dataset is taken from the IEEE Static Hand Gestures Dataset which is available on the IEEE dataport. It contains images in a png format which were initially captured in a 1920x1080 format and then formatted into a 400x400 grid to make it simpler for any future processing This is an open access dataset and is a part of the IEEE dataport and is thus a standardised dataset. Thus, it would be unchanged for anyone who does any processing on the same dataset and it would allow us to make comparisons easier.

IV. Practical Implementation

A. Hardware and Software Requirements

The code for the project was executed on a google colab notebook with the code being written entirely in python. From the google colab notebook, access to a GPU was obtained for the purposes of training the neural network in an efficient manner.

Parameters	Google Colab Specification
Disk Space	25GB
CPU Model	Intel Xeon
CPU Freq.	2.30 GHz
CPU Cores	2
Available RAM	25GB
GPU Model	NVidia GeForce RTX 1060
Available VRAM	4GB

B. Proposed Model

The implementation pipline contains an intial computer vision based module. This module makes use of the YOLO algorithm described above in order to identify and seperate the hand from the rest of the capture. Once the hand has been positively identified, it is sent to another module which performs operations on its HSV to convert it into greyscale. This pipeline is used in the final model after training in on the dataset described above.

The MobileNet model has been implemented in python using tensorflow and keras. Since it is a neural network based model, it is primarily developed using the keras submodule. Each layer has been defined specifically to work with the image data being fed into it. It has 28 layers including ReLU, convolutional and padding layers. This model was also trained with an ADAM optimizer and a categorical cross-entropy loss function.

The training was done on a GPU whose specifications were described above. Due to limited resources, the images had to be compressed before sending into the pipeline which could have had an impact on the results.

V. Output

A. Results

The loss and accuracy during the training for the MobileNet is shown below

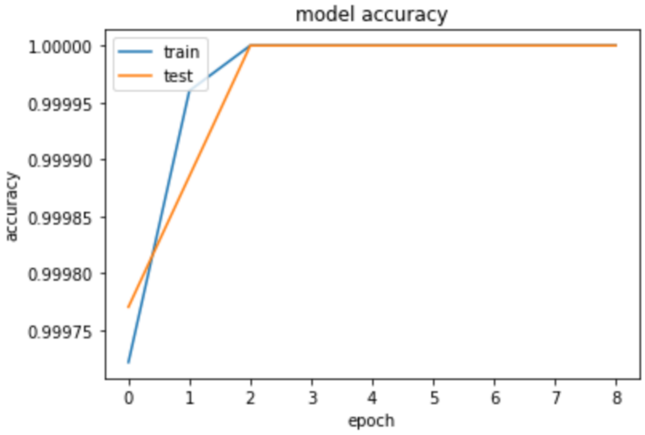


Fig. 3. Accuracy during training for the model

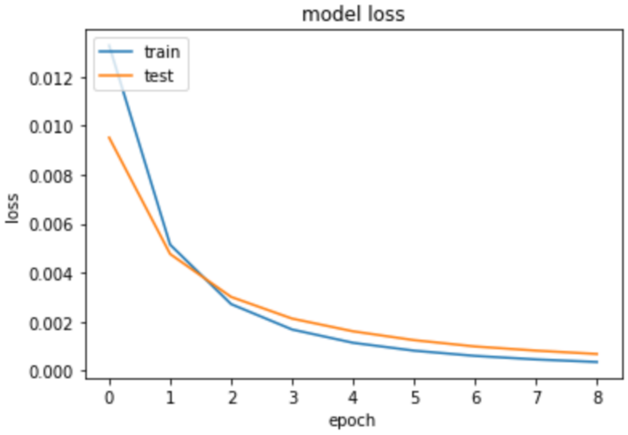


Fig. 4. The loss calculated during training

We then move on to plot the confusion matrix over the test data to see the classification metrics

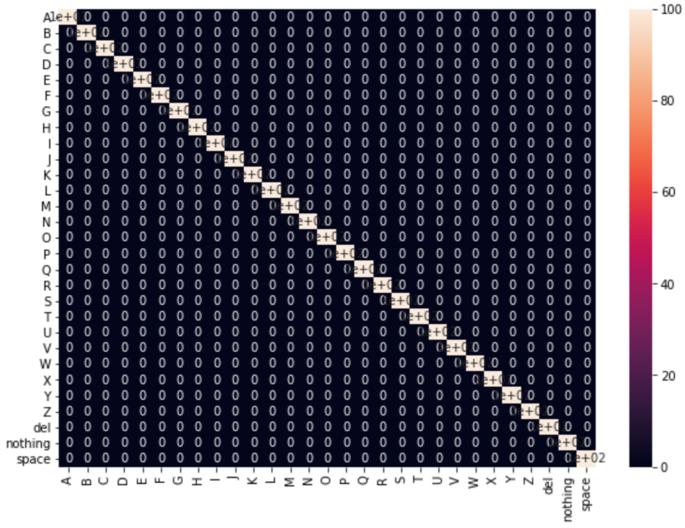


Fig. 5. Confusion Matrix for MobileNet

The confusion matrix shows us that the classifier trained on the data performs very well on the test set. There is very minor error and it also seems to generalize, but there is also a little risk of this result being overfitting. The reasons for this are discussed below in the interpretation.

B. Interpretation

From the confusion matrices shown above and from the was we can see the change in the model loss and accuracy, the MobileNet achoeves a high level of accuracy and is able to generalize on the test and training data. Some part of this might be due to this network being developed specifically for this task, being a modification of MobileNet which is a general framework, but we believe that this level of accuracy presents a significant step forward.

The system however is still overfitting a little and can be improved by fine-tuning the hyper-parameters to make the model generalise better on the test data. The main issue is that all the hend gestures in the dataset contain hands which are all from the same person. This caused the model to learn certain features which are unique only to those hands but not to any others. This causes a problem in generalization.

VI. Conclusion and Future Scope

In this paper we presented a method to classify sign language hand gestures. The model was based on the MobileNet architecture so that it could be deployed to any device and run there with minimal problems. The extended pipeline including YOLO and computer vision was also developed with the same task in mind.

One main issue with the dataset has been already discussed. It could help if any solution for that is produced. One solution to this would be to create a dataset which has hand gestures from different hands in order to ensure that the model does not unintentionally learn things that it is not supposed to.

In the future, a more robust system can be developed using the method described here on a system which has sufficient computing power to enable more hidden layers and more training data. This would help that system to generalise on any given input data and help in solving this open problem of sign language recognition.

References

[1] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, “Deep convolution neural network for image recognition,” *Ecological Informatics*, vol. 48, p. 257–268, 2018.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.

[3] A. Sahoo, G. Mishra, and K. Ravulakollu, “Sign language recognition: State of the art,” *ARPN Journal of Engineering and Applied Sciences*, vol. 9, pp. 116–134, 02 2014.

[4] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, p. 1285–1298, 2016.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 2017.

[6] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, “Sign language recognition using convolutional neural networks,” *Computer Vision - ECCV 2014 Workshops Lecture Notes in Computer Science*, p. 572–578, 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” 2016.

[8] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, “Scalable object detection using deep neural networks,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[9] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, “Object detection with deep learning: A review,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, p. 3212–3232, 2019.