ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 5, No. 6, 481-488 (2020)
www.astesj.com
Special Issue on Multidisciplinary Innovation in Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

American Sign Language Recognition Based on MobileNetV2

Kin Yun Lum*, Yeh Huann Goh, Yi Bin Lee

Department of Mechanical Engineering, Faculty of Engineering and Technology, Tunku Abdul Rahman University College, Kuala Lumpur, 43200, Malaysia

ARTICLE INFO

Article history:
Received: 30 August, 2020
Accepted: 20 October, 2020
Online: 20 November, 2020

Keywords:
Sign Language
Convolutional Neural Network
Depthwise Separable Convolution
MobileNet
Transfer Learning

ABSTRACT

Sign language is a form of communication language designed to link a deaf-mute person to the world. To express an idea it requires the use of hand gestures and body movement. However, the bulk of the general population remain uneducated to understand the sign language. Therefore, a translator is required to facilitate the communication. This paper wishes to extend the previously proposed Convolutional Neural Network (CNN) model for predicting American Sign Language with a MobileNetV2-based transfer learning model. The latter model effectively generalized on a dataset which is around 18 times larger with 5 additional groups of hand signs. Over 98% of the recognition accuracy had been reported. Because of its relatively fewer parameters and less intensive computational operations compared to other deep learning architectures, the model was also ideal to be implemented on mobile devices. The model will serve as the key to deploying a sign language translator software on smartphone to enhance communication efficiency between the deaf-mute person and the general public.

1 Introduction

This paper is an extension of work originally presented in conference IEEE International Conference on Signal and Image Processing Applications (ICSIPA) 2019 [1]. The work was enhanced in the following aspects:

1. Dataset: Previous study used only 4800 images with 24 groups of self-collected images except for letters J and Z, since both signs require movement. Furthermore, all images were taken from a fixed distance, which is not generally the case when it comes to actual application. The sign performer can appear at different distances from the smartphone camera. In this paper Kaggle's dataset was used. Each sign has variations in capturing distance and brightness. There are a total of 87,000 images with 29 classes. In addition to the 26 classes belonging to letter A-Z, there are an additional 3 classes for SPACE, DELETE and NOTHING. As per the description of the dataset, these 3 classes are found to be useful in real time application and classification.

2. Method: Previous proposed shallow network failed to generalize well with this new ~18x larger dataset. Instead, transfer learning using MobileNetV2 was implemented. Unlike other deep learning models, the MobileNetV2 gives high prediction accuracy without penalizing too much on computational cost and memory. This met our original intention to have this model deploy as a mobile application in the future.

Spoken language binds a significant section of the population. The spoken language would not, however, support the deaf and mute population. Statistics from the World Health Organization (WHO) shows there are at least 5% or 466 million of people suffer from hearing impairment [1]. An individual with hearing impairment may have a distorted speech or may not speak at all. This creates a barrier to contact between them and the society.

Fortunately, sign language comes to help for this special group of people. It is a kind of visual communication language that uses a combination of hand gesture, facial expression and body posture. It helps the hearing-impaired communities to express their feelings and addresses the general communication issues within their communities.

The sign language is completely different from spoken language. It has its own syntax and the way it expresses itself. For the general population this may be difficult to understand and practice [2].

*Corresponding Author: Kin Yun Lum, Department of Mechanical Engineering, Faculty of Engineering and Technology, Tunku Abdul Rahman University College, Kuala Lumpur, Malaysia, lumky@tarc.edu.my

Moreover, most general public are illiterate to sign language. Unless they are surrounded by a deaf-mute person, most appear to be uninterested and disregard the meaning of sign language. This presents a real barrier of contact between the deaf-mute community and the rest of society, as a issue which has yet to be completely resolved.

Innovations of technology in the area of sign language recognition try to break down this communication barrier. To date, it addresses the issues mainly with two different approaches: 1) Sensor based system 2) Vision based system [3].

The sensor based system is a wearable device which utilizes various sensors to capture motion, spatial positions and velocity of the hand. For example, cyber glove is a wearable device that employs flex sensor to detect fingers movements and Inertial Measurement Unit (IMU) sensor to track the hand motions [4]. There are also invasive and non-invasive electromyography (EMG) wearable sensors system to measure human muscle's electrical pulses and collect the bio-signal to detect fingers movements [5, 6]. These wearable system can be costly, cumbersome, inconvenient, heavy and uncomfortable for daily use.

Vision based system has recently become increasingly popular because there is no need to attach sensors physically to the human. There are attempts to use Leap Motion Controller [2] and Microsoft Kinect which have depth sensor [4, 7, 8] to construct the sign language recognition system. However, such technology is not widely accessible and it can be expensive to the users. Furthermore, the systems were implemented using desktop or laptop computer, which is impractical because of its weight and size.

Since hand sign gestures are perceived through vision, computer vision can be a subject of profound interest. One of the solutions is to build a vision based sign language recognition system with the commonly available smartphone camera. Vision-based systems are historically prone to reliability issues. Background noise, colours, and lighting are greatly varied under actual environment. This would result in a lower rate of detection. The latest advance in computer vision and machine learning technologies has shown considerable progress in the classification of images [9, 10]. In the competition for image classification, the algorithm called Convolutional Neural Networks (CNNs) and its derivative networks such as AlexNet, GoogleNet and ResNet achieved more than 90% accuracy [11]. Conventional machine learning relies on manual features extraction. Instead of creating complex handcrafted features, CNNs will automate the feature construction process. That would significantly reduce human error and increase the accuracy of detection.

There are several studies using CNN for the identification of American Sign Language (ASL) but the findings are not satisfactory [12]. Furthermore, while some deep networks may offer high classification accuracy, a large number of parameters and heavy mathematical operations limit their application in smartphones.

Our aim is to develop a computer vision system that can use mobile camera to recognize sign language, so that it can be used everywhere and anywhere. Previous attempt to create a CNN network to recognize 24 classes of sign language has demonstrated a detection accuracy of 95% experimentally. This paper extends the work to draw on a much larger dataset in order to enable greater generalization. The larger dataset contains 5 more classes than the previously used dataset. It included all 26 alphabets and another 3 classes for the sign of SPACE, DELETE and NOTHING to make the whole sign language model more complete. In addition, this dataset has also images which were captured at different distance from the camera, which may reflect a real-life application more closely. Transfer learning using MobileNetV2 was implemented to further improve the model accuracy to more than 98%.

The remainder of this paper is structured as follows: the relevant works will be discussed in Section 2. Followed by the section 3 concept and theory that will address the different CNN architectures and put emphasis on MobileNet, which is the transfer learning model to be implemented in this paper. Dataset description and preprocessing will be described in the methodology in Section 4. Then we will detail our model architecture and setup of experiments. The section ends by showing some evaluation metrics that we used to measure the performance of our model. Section 5 addresses experimental findings with discussions. Lastly, Section 6 ends with the conclusions of this paper with some suggestions for future study.

2 Related Works

The works related to vision based sign language recognition system can be categorized into two, which are non-CNN and CNN approaches. The non-CNN approaches were mainly the traditional methods which involved manual feature extraction and classification. The involved classification algorithms were k-nearest neighbors (KNN) and Neural Network [13, 14]. These approaches normally had lower reported prediction accuracy. On the other hands, the CNN approaches that had better reported results were either proposing their custom CNN models [7, 8, 15] or implementing transfer learning [11, 12, 16, 17]. There was also literature which compared both their custom CNN model and transfer learning [18].

For custom CNN models, Vivek Bheda et. al. used a deep convolutional network to classify ASL with alphabets and digits. The proposed network had 3 cascaded convolutional layers and 1 max-pooling layer. There are two hidden layers with dropout before connecting to the output layer. Accuracy of 82.5% had been reported [7].

Ameen. S. et. al. separated the image and depth information from Microsoft Kinect input and used two convolution layers to perform feature extraction on each of them. Both extracted feature maps were combined in the second stage of convolution before passing to the classifier. The reported precision and recall were 82% and 80% respectively [15].

Similar architecture was used by Pigou. L. et. al. to extract hand features and upper body features separately from Microsoft Kinect dataset. Each CNN was three layers in depth. They had reported an accuracy of 91.7% [8].

For transfer learning, Das, A. et. al. used Inception v3 on custom processed static gesture images and obtained an average validation accuracy of 90% [16]. Alashhab. S. et. al. tested on VGG16, VGG19, ResNet, Xception, Inception V3, MobileNet and SqueezeNet. They were dealing with hand gestures detection instead of sign language. There were 5 classes of hand gestures used. The highest accuracy of transfer learning usign MobileNet was 99.45% [11]. Kania, K.et. al. build a network to recognize ASL with reference to the Wide Residual Networks. Data augmentation was used to increase the number of training sets and reported a high-

est accuracy of 93.3% [12]. Garcia and Viesca used GoogLeNet architecture to recognize 24 classes of sign language with only 70% or reported accuracy [17]. As far as our literature, there was only one work to compare custom CNN model and transfer learning done by Bousbai, K. and Merah, M. Their training set had 1815 coloured images with black background collected from five volunteers. Their custom CNN had a reported efficiency of 98.9%, slightly surpassed the transfer learning model with 97.06% accuracy [18].

In general summary, transfer learning can perform better than most of the custom CNN model when come to pure image recognition. Custom model is only needed usually to cater for the extra input like depth information. Transfer learning model specially designed based on pure images dataset like ImageNet might not doing as good as the custom model. Other than this reason, transfer learning is a more appropriate solution to develop a well performing model. Therefore, it is being used in this paper to enhance the results of our previous work.

3 Concept and Theory

3.1 Convolutional Neural Network (CNN) Architecture

Convolutional Neural Network (CNN) has proven to be very effective to handle image classification problems. It is in use by many industry leaders such as Amazon, Google and Facebook [8]. The convolution layers in the CNN perform multiple discrete convolutions on the input with defined number of trainable weights filters. The filters' weights were updated during training. After applying filtering on each channel, it will capture the image features such as lines, edges, colors and other visual elements. The deeper the network goes, more complex features like shapes and patterns can be generated. Combining all these features together with the activation functions can produce feature maps which can then pass to the ordinary neural network classifier to correctly classify the object.

Due to significant progress has been achieved in image classification, there were various CNN architectures introduced such as VGG16, ResNet and Inception Net. These models were trained and verified on large-scale annotated datasets like ImageNet and proven to be effective. Instead of constructing the CNN from scratch, leverage on the transfer learning on these architectures can significantly reduce the time required to produce a working model for a particular problem. Moreover, since these models were trained on ImageNet which consists of 1.2 million categorized natural images of 1000+ classes. Using the trained weights as a starting point will definitely help the model to converge faster as compared to training from scratch. As discussed by Shin, H. C. et. al, even though there is a disparity between the ImageNet and our sign language images, the deep architecture which is trained comprehensively can as well generalize on other specific domain problem with only minor fine-tuning training [19]. Among different CNN architectures, MobileNet was employed in this paper because it is an efficient network architecture designed to run on any computationally limited platform.

3.2 MobileNetV1

MobileNet had been introduced as a small networks and it is optimized for latency to match the restricted resource application such as in mobile devices, robotics and self-driving vehicles. These application platforms are not as powerful as a general purpose computer, but still requires comparable recognition accuracy under a timely computation.

Unlike other network architectures use shrinking or compression to reduce its size and in return to improve on computational costs, MobileNet was completely rebuilt using depthwise separable convolutions. Depthwise separable convolution is a form of factorized convolutions which decomposes a conventional convolution into a depthwise convolution and a pointwise convolutoin. This factorization ends up reducing computation cost and model size drastically. The rationale behind this is to reduce the costly convolution process due to multiple multiplication as explained below [20].

Examine a standard convolution process, let an input image, F of dimension $D_W \times D_H \times M$ where D_W and D_H correspond to the spatial width and height of the input image and M is the number of input channels (depth). There are N numbers of square filters, K with size of $D_K \times D_K \times M$ where $D_K$is the spatial dimension of the kernel. An example of standard convolution with an input matrix with channel, $M = 3$ and N number of 3×3 kernels is illustrated in Figure 1a. All the convolution processes discussed in this section are assumed to have stride of one and padding. This standard convolution between F and N numbers of K will have a computational cost of:

$$J_c = N\left[(D_W \times D_H) \times (D_K \times D_K \times M)\right] \tag{1}$$

The MobileNet addresses this issue by splitting the convolution process into two steps, namely depthwise convolution and pointwise convolution. The whole process is called depthwise separable convolution. In the depthwise convolution process, instead of using kernel with depth M, single layer kernels are applied on each input channel as showed in Figure 1b. This results in a computational cost of:

$$J_d = M\left[(D_W \times D_H) \times (D_K \times D_K)\right] \tag{2}$$

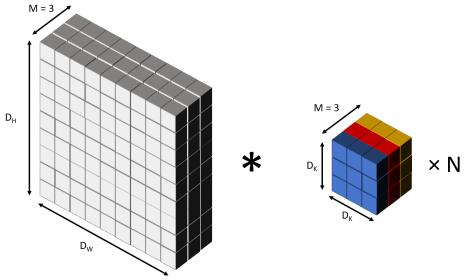
Then, pointwise convolution is performed on the output of the above depthwise convolution feature map. Pointwise convolution is being done by N number of 1×1 kernels with depth of M. This process is also called linear combination since the output matrix is of summation of linear scaling multiples. An pictorial illustration of this process is shown in Figure 1c. The computational cost of the pointwise convolution is:

$$J_p = N\left(M \times D_W \times D_H\right) \tag{3}$$

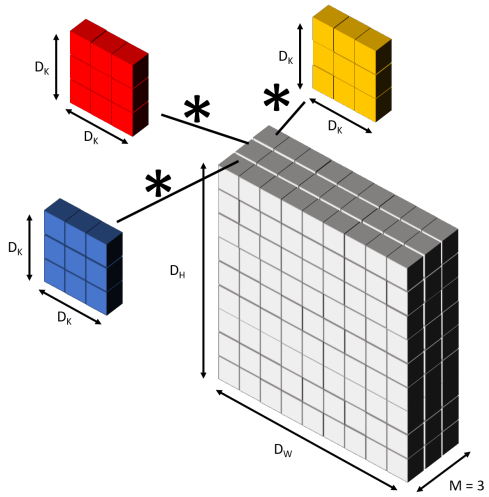
The total computational cost of a complete depthwise separable convolution is $J_{dp} = J_d + J_p$. Comparing to the standard convolution, there is a reduction factor of:

$$J_r = \frac{J_{dp}}{J_c} = \frac{1}{N} + \frac{1}{D_K^2} \tag{4}$$

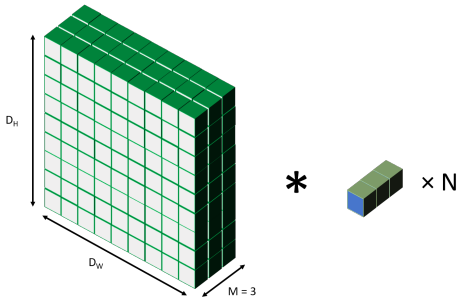
For a depthwise separable convolution with 3×3 kernels used, the computational cost can be 8 to 9 times lesser with only minor trade-off in accuracy [20].



(a) Standard convolution



(b) Depthwise convolution



(c) Pointwise convolution

Figure 1: Comparison between (a) Standard convolutoin (b) Depthwise convolution and (c) Pointwise convolution

3.3 MobileNetV2

MobileNetV2 is an improved version of MobileNetV1. MobileNetV1 or most deep convolution network uses the Rectified Linear Unit (ReLU) activation function defined by

$$f(x) = \begin{cases} x, & \text{if } x \geqslant 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{5}$$

The ReLU activation function disregards any values smaller than 0. This non-linear transformation is argued to result in the loss of information, especially on input with a lower number of channels. ReLU may have less impact on input with lots of channels because the missing information in particular activation might still be preserved by other channels [10].

To tackle this issue, MobileNetV2 uses a narrow -> wide -> narrow approach in its inverted residual block. The low-dimensional input is first expanded by using a pointwise 1×1 convolution to produce a higher dimensional space which can cater for the information loss caused by the ReLU activation. Then, spatial filtering using depthwise convolution with ReLU activation was implemented in this higher dimensional feature map. Finally, another pointwise convolution was performed to project back to a lower dimensional output feature map. For the last step, linear activation instead of ReLU was used to preserve more information when encoding to a lower dimensional output map. This idea was called linear bottleneck by the authors. The non-linear transformation only happened in the internal expanded higher dimensional space inside the block. When come to lower dimensional output, linear transformation was used [10].

Besides, a skip connection similar to residual block is added between the input and output of the inverted residual block to allow gradient flow during backpropagation. This approach is essential to build a deep network. One last minor adjustment is the use of ReLU6 in this inverted residual block which capped the maximum output to 6 as defined by:

$$f(x) = min(max(0, x), 6) \tag{6}$$

The complete inverted residual block discussed in MobileNetV2 is shown by Figure 2.

4 Methodology

4.1 Dataset

Figure 3 shows the American Sign Language (ASL) for all alphabets. In the previous article, dataset was produced by smartphone camera captures [1]. There were a collection of 4800 images with 24 classes of gestures. The signs for character J and Z were excluded due to dynamic. All the images were taken from at a fixed camera distance with little lighting variation as displayed in Figure 4.

The challenge of developing a usable sign language recognition model lies in the variation in size, position, and shapes of the input image [15]. In real life applications, the sign language performer might stands at different distance relative to the smartphone camera. Provided data augmentation which is widely adopted in training CNN with limited dataset, which introduces more variations with scaling, rotating, flipping, shifting, shearing, etc. However, this image augmentation failed to reproduce the realistic perspective variations as discussed by Wenjin Tao et. al [4].

To extend the work, the dataset from Kaggle named “ASL Alphabet” was used [21] . Each sign has variations in distance capture and illumination. An example of sign B was shown in Figure 5. There are a total of 87,000 images in 29 classes. Besides the 26 classes belong to the letter A - Z, there are three additional classes for SPACE, DELETE and NOTHING. As per dataset description, these 3 classes are found to be helpful in real time application and classification.

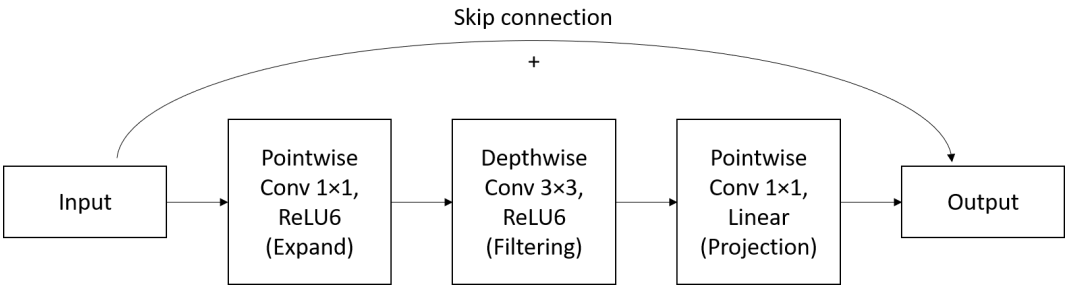


Figure 2: Inverted residual block

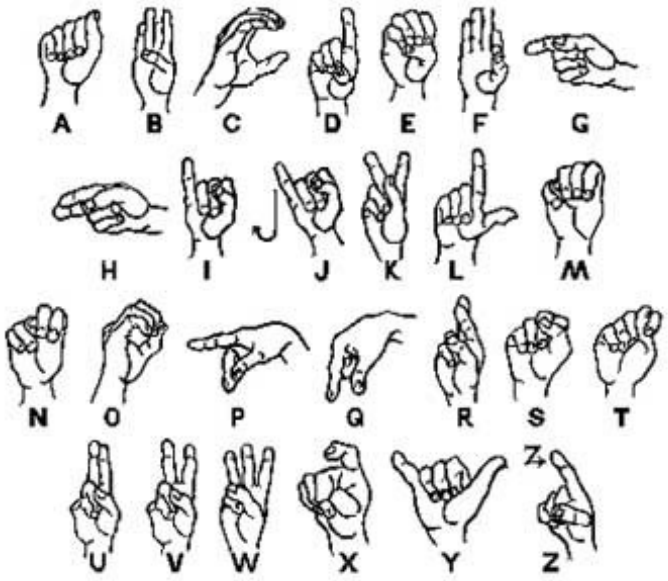


Figure 3: American Sign Language for all alphabets

Figure 4: Example self-captured dataset from previous article. From left to right (top) A, B, C and (bottom) D, E, F [1]

4.2 Model Architecture

The new dataset in this paper is approximately 18 times larger than the previous dataset. The previous proposed CNN network which had only two cascaded convolution layers failed to generalize to this dataset. The model was therefore being reconstructed using transfer learning with MobileNetV2. This CNN architecture was optimized for mobile devices to give high prediction accuracy while keeping the parameters and mathematical operations low. Compared to MobileNetV1, the MobileNetV2 is about 30 - 40% faster when tested on Google Pixel phone due to 2 times fewer mathematical operation and 30% lesser parameters. Not only that, it outperforms MobileNetV1 in terms of recognition accuracy [18].

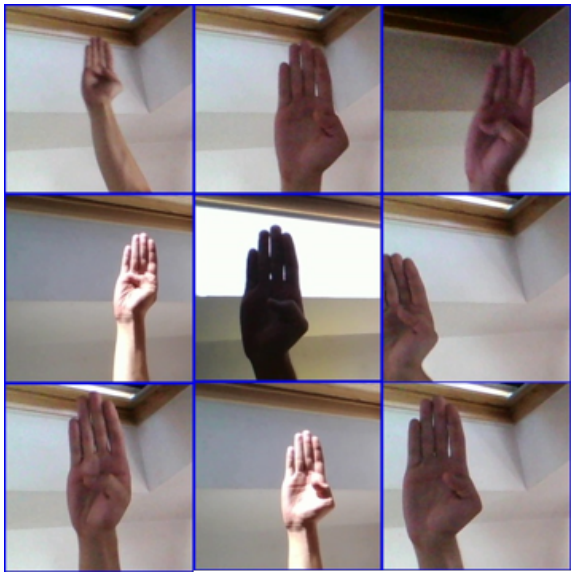


Figure 5: Example sign for letter 'B' under different capturing distance and lighting condition

All 29 classes had been used in this work. The images were resized to 224×224 pixels as suggested in the article by Samer Alashhab. et. al. [11]. Among different input sizes, 224×224 pixels gives best performance by experiment. The inputs were shuffled randomly and normalized. The images were divided into training and validation sets using a ratio of 80:20. The training set has 69,600 images while the validation set has 17,400 images. In view of the larger dataset, no data augmentation was carried out.

The built architecture is shown in Figure 6. The MobileNetV2 was preloaded with weights trained by ImageNet. Average pooling layer was added to the MobileNetV2 output feature map. Continued with a dense neural network layer of 1000 neurons. The activation function for this dense network was standard ReLU. A 50% dropout layer was added after that to prevent overfitting. Final layer was the dense layer that provides classification for 29 classes of images. It had a softmax activation function as defined by:

$$\sigma(z)_j = \frac{e_j^z}{\sum_{k=1}^{K} e_k^z} \tag{7}$$

The model was compiled using categorical crossentropy loss function. The model had about 3.5 millions training parameters. The optimizer used was RMSprop with a learning rate of 1×10^{-4}. Since the sign language images are differ from the ImageNet dataset, all the layers were unfreezed and retrained. Experiments were all conducted on a computer with Intel Core i5 CPU, 12 GB RAM and a NVIDIA GeForce GTX 1060. All of the steps mentioned in this section were performed using Keras deep learning framework with TensorFlow as the backend.

4.3 Performance Metrics

The performance metrics used to evaluate the build model are defined below, where TP stands for True Positive (correct prediction), TN is True Negative (correct rejection), FP is denoted as False Positive (incorrect prediction) and FN is defined as False Negative (incorrect rejection).

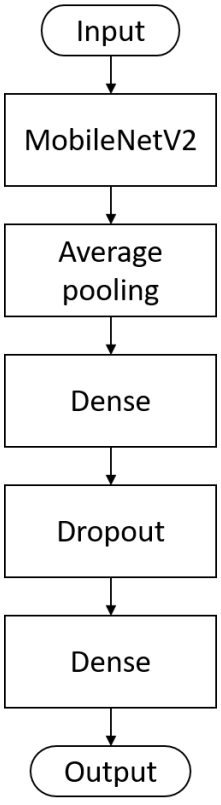


Figure 6: Model architecture

1. Accuracy - the overall correct prediction of the model

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

2. Precision - ratio of correct prediction over all the predicted values of a particular class.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

3. Recall - ratio of correct prediction over all predicted values that are supposed to be positive.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

4. F1-Score - weighted average of precision and recall. It tells the balance between precision and recall.

$$F1\ score = 2\left[\frac{Precision \times Recall}{Precision + Recall}\right] \tag{11}$$

5 Results and Discussion

The highest overall accuracy obtained by this model was 98.67%. The confusion matrix is shown in Figure 7. For each of the sign, 28 out of 29 were recorded with accuracy higher than 95%. The highest single class accuracy was 100% and the lowest was 85.67%. Lowest accuracy was recorded for the letter Q. There were only 514 out of 600 images were correctly predicted. The rests were all being predicted as P. This was believed to be the similarity existed in these two signs, where the index finger and thumb are both pointed out.

Looking at the precision, recall, and f1-score for each class as shown in Table 1, most of them obtained a score of 95% and above. The precision of letter P and recall of letter Q was lower due to the above mentioned similarity. Hence it also lowered the f1-score for these two classes. The second lowest recall was the sign for letter X which scored 92.83%. Some of them were being wrongly classified as either S or U due to the similar rounded fist shape.

Table 2 shows the comparison with some previous works. To have a fair comparison, only those works implementing transfer learning were compared. In the comparision, there was only one study done by Alashhab. S. et. al. had slightly better accuracy then this work, but they were not working with ASL and there were only 5 classes of hand signs used. Other than that, this work showed better results than the prior works. For a similar MobileNetV2 transfer learning architecture for ASL, there is also a marginally improvement over the work did by Bousbai, K and Merah, M.

Table 1: Precision, Recall, F1-score for each class

	Precision	Recall	F1-score
A	0.9677	1.0000	0.9836
B	0.9933	0.9917	0.9925
C	1.0000	1.0000	1.0000
D	1.0000	1.0000	1.0000
E	0.9693	1.0000	0.9844
F	0.9950	1.0000	0.9975
G	1.0000	0.9850	0.9924
H	0.9709	1.0000	0.9852
I	1.0000	0.9800	0.9899
J	1.0000	0.9633	0.9813
K	1.0000	1.0000	1.0000
L	1.0000	1.0000	1.0000
M	0.9771	0.9950	0.9860
N	0.9817	0.9850	0.9834
O	1.0000	0.9950	0.9975
P	0.8721	0.9883	0.9266
Q	1.0000	0.8567	0.9228
R	0.9934	1.0000	0.9967
S	0.9769	0.9850	0.9809
T	0.9983	0.9783	0.9882
U	0.9707	0.9933	0.9819
V	1.0000	0.9917	0.9958
W	0.9868	1.0000	0.9934
X	0.9893	0.9283	0.9579
Y	0.9983	1.0000	0.9992
Z	0.9983	0.9983	0.9983
delete	0.9967	1.0000	0.9983
nothing	1.0000	1.0000	1.0000
space	1.0000	1.0000	1.0000

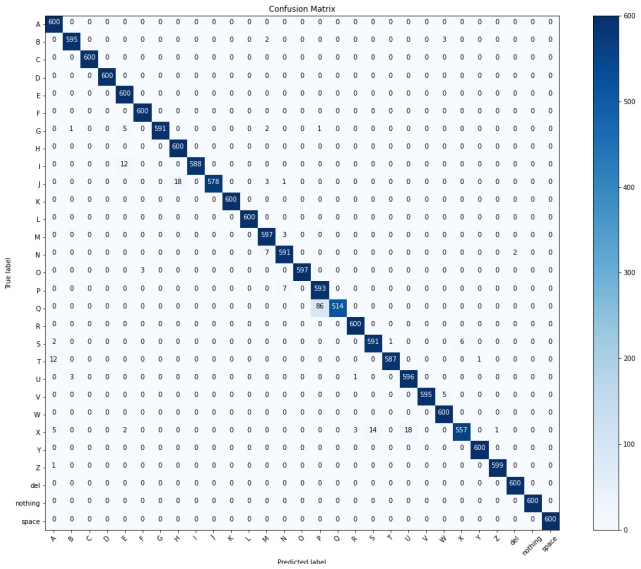


Figure 7: Confusion Matrix

Table 2: Comparison of this work and previous works

Authors	**Description**	**Accuracy**
Das, A. et. al., 2018	Transfer learning using Inception v3 on custom dataset	90.0%
Alashhab, S. et. al., 2018	Transfer learning on multiple architecture like VGGNet, ResNet, etc. on custom 5 classes of hand gestures.	99.45% (MobileNet)
Kania, K.et. al., 2018	Transfer learning using Wide Residual Networks with data augmentation on ASLs	93.3%
Garcia and Viesca, 2016	CNN on 24 ASLs with GoogLeNet tranfer learning	70%
Bousbai, K. and Merah, M., 2019	Compare custom CNN model and transfer learning using MobileNetV2 on ASLs	97.06% (MobileNetV2)
Our previous work, 2019	**Custom CNN on 24 ASLs using phone camera**	**95%**
This work	**Transfer learning using MobileNetV2 on 29 classes of ASLs**	**98.67%**

6 Conclusion and Future Work

This paper presented a model that was able to recognize American Sign Language with 98.67% accuracy. It is an extension of the work originally presented in ICSIPA 2019. Besides the improvement of the model accuracy, the following additional highlights are listed:

1. A much bigger dataset was used with additional 5 classes to make the ASL more complete. That allowed a better generalization of the model. In addition, this distance varying images given in the dataset could also make the model less sensitive to the difference in distance between the ASL users and the camera.

2. Even with the deep transfer learning network used, we still maintained our original intention to make the model suitable for running on mobile devices in order to use the smartphone camera for the convenience of users.

The model was less computational burdening since it had fewer parameters and lesser mathematical computations. With the implementation of the model as an smartphone app, it is believed to be able to improve the quality of communication between the deaf-mute person and the general public.

To offer more usability, this work can be continued with real-time video based sign language recognition. To reduce the effects of ambient noise, there is a need for real-time video processing, such as region of interest segmentation and hand tracking. Apart from that, the image occlusion has yet to be investigated. This can be a challenging issue when part of the performing signs is blocked.

Acknowledgment I wish to acknowledge Tunku Abdul Rahman University College for sponsoring this research.

References

[1] L. Y. Bin, G. Y. Huann, L. K. Yun, "Study of Convolutional Neural Network in Recognizing Static American Sign Language," in 2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 41–45, IEEE, 2019, doi:10.1109/ICSIPA45851.2019.8977767.

[2] T.W. Chong, B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," Sensors, **18**(10), 3554, 2018, doi:10.3390/s18103554.

[3] M. A. Jalal, R. Chen, R. K. Moore, L. Mihaylova, "American sign language posture understanding with deep neural networks," in 2018 21st International Conference on Information Fusion (FUSION), 573–579, IEEE, 2018, doi: 10.23919/ICIF.2018.8455725.

[4] W. Tao, M. C. Leu, Z. Yin, "American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion," Engineering Applications of Artificial Intelligence, **76**, 202–213, 2018, doi:10.1016/j.engappai.2018.09.006.

[5] M. J. Cheok, Z. Omar, M. H. Jaward, "A review of hand gesture and sign language recognition techniques," International Journal of Machine Learning and Cybernetics, **10**(1), 131–153, 2019, doi:10.1007/s13042-017-0705-5.

[6] J. Wu, L. Sun, R. Jafari, "A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors," IEEE journal of biomedical and health informatics, **20**(5), 1281–1290, 2016, doi: 10.1109/JBHI.2016.2598302.

[7] V. Bheda, D. Radpour, "Using deep convolutional networks for gesture recognition in American sign language," arXiv preprint arXiv:1710.06836, 2017.

[8] L. Pigou, S. Dieleman, P.-J. Kindermans, B. Schrauwen, "Sign language recognition using convolutional neural networks," in European Conference on Computer Vision, 572–578, Springer, 2014, doi:10.1007/978-3-319-16178-5_40.

[9] P. Molchanov, S. Gupta, K. Kim, J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 1–7, 2015, doi: 10.1109/CVPRW.2015.7301342.

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.

[11] S. Alashhab, A.-J. Gallego, M. Á. Lozano, "Hand gesture detection with convolutional neural networks," in International Symposium on Distributed Computing and Artificial Intelligence, 45–52, Springer, 2018, doi:10.1007/978-3-319-94649-8_6.

[12] K. Kania, U. Markowska-Kaczmar, "American Sign Language Fingerspelling Recognition Using Wide Residual Networks," in International Conference on Artificial Intelligence and Soft Computing, 97–107, Springer, 2018, doi: 10.1007/978-3-319-91253-0_10.

[13] R. Hartanto, A. Kartikasari, "Android based real-time static Indonesian sign language recognition system prototype," in 2016 8th International Conference on information Technology and Electrical Engineering (ICITEE), 1–6, IEEE, 2016, doi:10.1109/ICITEED.2016.7863311.

[14] H. M. Ewald, I. Patil, S. Ranmuthu, "ASL fingerspelling Interpretation," University of Stanford, Reports, 2016.

[15] S. Ameen, S. Vadera, "A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images," Expert Systems, **34**(3), e12197, 2017, doi:10.1111/exsy.12197.

[16] A. Das, S. Gawde, K. Suratwala, D. Kalbande, "Sign language recognition using deep learning on custom processed static gesture images," in 2018 International Conference on Smart City and Emerging Technology (ICSCET), 1–6, IEEE, 2018, doi:10.1109/ICSCET.2018.8537248.

[17] B. Garcia, S. A. Viesca, "Real-time American sign language recognition with convolutional neural networks," Convolutional Neural Networks for Visual Recognition, **2**, 225–232, 2016.

[18] K. Bousbai, M. Merah, "A Comparative Study of Hand Gestures Recognition Based on MobileNetV2 and ConvNet Models," in 2019 6th International Conference on Image and Signal Processing and their Applications (ISPA), 1–6, IEEE, 2019, doi:10.1109/ISPA48434.2019.8966918.

[19] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, **35**(5), 1285–1298, 2016, doi: 10.1109/TMI.2016.2528162.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint arXiv:1704.04861, 2017.

[21] Akash, "ASL Alphabet, https://www.kaggle.com/grassknoted/asl-alphabet," 2018 (Accessed July 1, 2020).