Performance Evaluation of Resnet Model on Sign Language Recognition

Millicent Agangiba
University of Mines and Technology
Tarkwa, Ghana

Ezekiel M. Martey
University of Mines and Technology
Tarkwa, Ghana

William A. Agangiba
University of Mines and Technology
Tarkwa, Ghana

Obed Appiah
University of Energy and Natural Resources
Sunyani, Ghana

ABSTRACT
Communication is an important tool for sharing one's ideas and thoughts and as such its role in our everyday lives cannot be over emphasised. Sign language is a form of communication used by the deaf and those hard-of-hearing. However, a challenge arises when deaf people have to communicate their ideas to those in the mainstream population. An automatic translator can be an effective way to address this problem. In this study, the performance of the ResNet model and its variants are evaluated on two different datasets. The first dataset contains images of American Sign language (ASL) data and the second dataset consists of images of Indian Sign language (ISL). The is a one-handed sign language, while ISL is mainly a two-handed sign language with complex shapes. ResNet variants such as Resnet18, ResNet34, ResNet50, ResNet101 and ResNet152 have been tested on these standard datasets. We conducted experiments by using deep neural networks to make recommendations and predictions in sign language. Experimental results using a standard dataset demonstrate that the model with 152 layers achieves the highest accuracy.

General Terms
Artificial Intelligence, Deep learning, Computer Vision.

Keywords
Deep Neural Network, ResNet, American Sign Language, Indian Sign Language, Image Recognition

1. INTRODUCTION
Disability is a major challenge that affects, around 15 percent of the total world's population, with the majority residing in developing countries [1-3]. Among the disability category, those with hearing and speech impairment are the most challenged in communicating with the population mainstream [4]. However, this population of our society are mostly overlooked because of the lack of communication as sign language is not understood by the majority hearing population. To address this problem, people with hearing and speech impairment rely on sign language to communicate.

Sign language communication combines hands, arm movements, body movements, and facial gestures to convey the speaker's feelings [5]. Japanese Sign Language (JSL), Bangla Sign Language (BdSL), British Sign Language (BSL), Adamorobe Sign Language (AdaSL), American Sign Language (ASL) and Indian Sign Language (ISL) are well-known sign languages in the world. Most communication between the deaf and the hearing community relies heavily on the services of a human translato[6]. Consequently, it becomes difficult for the deaf to access critical services such as health care independently. Also, services provided by the translators come at a fee thereby increasing their cost burden [7]. With the advent of automatic sign language recognition systems, signs can be translated into corresponding text or speech without the help of human translators. This type of system provides huge benefits to the deaf community through human-computer interaction.

Sign language can be categorised into single-handed (e.g. ASL) and double-handed (e.g. ISL). In contrast, gestures in single-handed sign language require one hand, while double-handed sign language requires two hands and has more complex shapes [8]. More so, all sign languages are independent and distinct.

Several technologies have been deployed in the past to improve the communication gap for deaf people, such as the introduction of vision-based and glove-based gesture translators, which capture hand movements and translate them into text, and finally output them as speech [9-11]. These recent methods mainly use machine learning techniques, nonetheless, a paradigm shift to deep learning methods has been reported lately. A significant advantage of deep learning is the effectiveness of its feature engineering [12]. Mannan et al[13] featured fine-tuned CNN architecture with several convolutional layers, dropout max-pooling, and dense layers for the recognition of sign language. The authors used the DeepCNN to predict ASL hand gestures alphabets on the Sign Language MNIST dataset. The results demonstrated that proposed model obtained a high recognition accuracy value of 99.67%.

A work by Lum et al [14] utilised transfer learning model based on MobileNetV2 for predicting American Sign Language. The model was implemented as smartphone sign language translatation software to improve public communication. The paper achieved a recognition accuracy of 98%. Agrawal et al [15] developed a hand recognition system based on CNN and image processing methods. A camera was used in the proposed system to create a real-time gesture feed. The Images were created from the video frames in the movie. The images were then preprocessed and fed into the system to train, recognize and convert sign language gestures into text and audio output. However, the system's performance was impacted by background changes.

The performance of deep learning methods can be enhanced by deeper networks to address complex vision problems. However, such networks are susceptible to the issues of vanishing gradient. This will not only lead to saturation of accuracy but also reduce training accuracy. A notable contribution by He et al.[16] used ResNet to resolve the problem in deep Convolutional Neural Networks (CNNs).

Inspired by ResNet, different variants of ResNet are proposed based on the number of layers. These layers range from 18 to 1202. ResNet 18 and ResNet34 are small networks while

ResNet50, 101, 152 etc are deep networks. The novel idea of ResNet is that it employs shortcut connections in the residual block to aid the gradient flow in the backpropagation step, to gain accuracy in the training phase of a very deep network.

Rathi et al [17], in their work, proposed a 2-level Deep Neural Network Architecture based on ResNet50 to categorise words written with the fingers. The Level 1 model partitions the input image into one of the four sets. The accompanying secondary model employs the set to which an image has been allocated as input to estimate the image's actual class. The approach yielded an accuracy of 99.03%.

Saleh and Issah [18] employed transfer learning to train deep convolutional neural networks to enhance the accuracy of gesture detection in 32 Arabic sign languages. The proposed methodology worked by creating models with structures similar to ResNet152 and VGG16. Each network's layers are loaded with the weights from the pre-trained models. The softmax classification layer is inserted as the final layer following the last completely connected layer. When given conventional 2D images in a variety of Arabic sign languages, these networks produced results with an accuracy of about 99%.

A recent experiment by Alleema and Chandrasekaran [19], presented a modified convolutional neural network with a residual 101 classifier to recognise ASL images. The Sign language identification system had three basic steps to accomplish the recognition task. The initial step is to preprocess the image to minimise noise. The second step is segmentation to remove feeble edges in the image. Lastly, classification is performed by means of a Modified CNN Deep Residual 101 classifier. Classified images correctly recognize American Sign Language. The result indicated that the proposed method generated an accuracy score of 95%.

Based on the ResNet, several modifications, novel architectures and designs are proposed for diverse image recognition tasks [16, 20-22]. Different kinds of the ResNet structures have been proposed, which are composed of different combinations of convolutional layers, activation functions [16], rectified linear unit [23] and batch normalization (BN)[24]. The ResNeXt developed by Xie et al. [25] is a redesigned version of the ResNet, using a split transformation and aggregation scheme for image recognition.

Generally, the above ResNet and its variants models have yielded promising results. To this end, this paper evaluates the ResNet model and its sub-models in terms of computational requirements and accuracy to propose the best ResNet architecture for sign language prediction and recognition.

2. RESOURCES AND METHOD USED

2.1 Description of Dataset

A concise description of the datasets used to train and validate the models is presented below.

American Sign Language (ASL), Kaggle [26]: This dataset has a total of 87, 000 images divided into 29 classes. Twenty-six (26) classes contain the letters A - Z and the remaining classes contain "space", "delete" and "nothing" respectively.

Indian Sign language (ISL), Kaggle [27]: This dataset consists of alphabet-numbers-related images. The ISL alphabet images were considered for this experiment. The total images amount to 62400 consisting of 26 classes of which 1200 are for each letter from A – Z. Figs. 1 and 2 show samples of images in the two datasets.

2.2 Data Pre-processing and Augmentation

Data pre-processing is done to transform the raw data into a standard form for further processing. To provide a reasonable sense of model effectiveness, the dataset is divided into training and validation sets [28]. 80% of the dataset was used as the training set and the remaining for the validation set. Mean subtraction and normalization were applied to each image in the training dataset. Mean subtraction involves subtracting the mean across every individual datapoint in the data and has the effect of zero-centering the data. The normalization process also involves the dividing of each zero-centered dimension by its standard deviation. The process has the effect of making the network converge faster. Equations 1 and 2 show the computation of both processes:

$$X^I = X - \bar{x} \quad (1)$$

where X^I is the normalized dataset, X is the original dataset and $\bar{x}$ is the mean of X.

$$X^{II} = \frac{X^I}{\sigma x} \quad (2)$$

where, X^{II} is the standardized dataset, and σx is the standard deviation of X.

The images were resized to a pixel size of 64 × 64 pixels and fed to the network as input. The resize was done to reduce the number of data points the network has to compute. Additionally, these transformations were randomly applied to the images in the training set: horizontal and vertical flips, lighting and contrast changes, rotation, zoom and symmetric wrap.

Fig. 1. American Sign Language

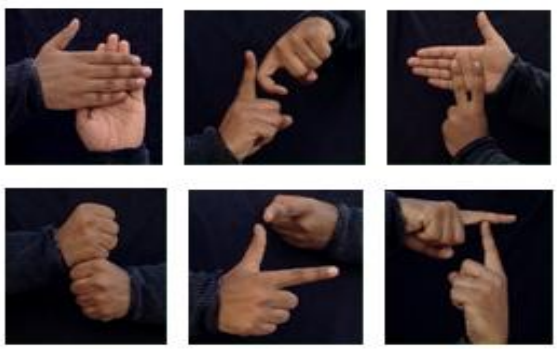

Fig. 2. Indian Sign language

2.3 Residual Network Model

To create an efficient model for neural networks, the network architecture needs to be carefully analysed before being

chosen. The variants of ResNet architecture ResNet18, ResNet34, ResNet50 ResNet101 and ResNet152 were employed as used in He et al [16]. Table 1 presents the architectures of the models used. ResNet model was chosen because it handles the vanishing gradient problem well by improving the flow of information and gradients through the network. This allows for much deeper neural layers without increasing the training error rate. A block diagram of the sign language recognition system is shown in Fig. 3.

Conventional deep neural networks perform computation using equation 3.

$$x_l = H_l(x_{l-1}) \quad (3)$$

The output of the $\ell\ th$ layer is fed as input to the $(\ell + 1)\ th$ layer [29]. ResNets [16] adds a skip-connection that bypasses the non-linear transformations with an identity function in equation 4:

$$x_l = H_l(x_{l-1}) + x_{l-1} \quad (4)$$

In the experiment, no extra parameters were introduced in the ResNet models. To classify the images in the two datasets, the final block is dropped and replaced with a fully connected dense layer with an input size that depends on the number of classes of the datasets, representing each of the classes to be classified.

Average pooling is applied to the end of this depth. This is done to collapse all extra channels to a 1D tensor at this depth. It is achieved by taking an average across the extra channels.

Softmax nonlinearity is used on the scores outputted by the dense layer. The softmax function computes final class scores that will be fed into the loss function or output during testing. These scores are the neural network's estimated probabilities for each class. The ith class probability (x) is computed using equation 5:

$$f(x)_i = \frac{e^{x_i}}{\sum e_{jx}} \quad (5)$$

The network's weights are initialized with weights from a model pre-trained on ImageNet. The network's knowledge of this pre-trained model is transferred to the task of identifying finger spellings. The model takes the fingerspelling image as input.

2.4 Training the Model

Backpropagation is used to update the parameters of the ResNets model [30]. Backpropagation uses the gradient of a loss function to decide how to update the network's parameters to minimize the loss.

2.4.1 Loss Function

The model aims to specifically optimize the loss function. The standard categorical cross-entropy loss was used in this project. It is computed from the outputs of the final softmax layer. When the softmax activation function is combined with the cross-entropy loss, they form the softmax loss. The softmax loss can be written for the ith input feature xi with a label yi using equation 6:

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i - log\left(\frac{e^{f_{yi}}}{\sum_j e^{f_j}}\right) \quad (5)$$

where N is the amount of training data and the jth element $j \in [1, K]$, K is the number of classes) of the vector of class scores, f is represented by fj.

2.4.2 Optimizer

The network was trained using AdamW [31 which tackles the issues of poor generalization with the popular Adam optimization algorithm [32]. Although Stochastic Gradient Descent (SGD) with momentum can be used to achieve state-of-the-art results for many tasks such as object recognition. [20] and machine translation. Wilson et al. [33] justify that, adaptive learning rate methods converge to different and less optimal minima than SGD with momentum.

2.4.3 Learning Rate Finder

The learning rate is a hyperparameter that determines how often the weights of the network are modified for the loss gradient. To avoid this rate from deviating from the minimum gradient, the technique proposed by Smith [34] is used. The learning rate is gradually increased after each mini-batch. The loss at each increment is recorded and plotted. Analysing the slope of the plot indicates the optimum learning rate, which is associated with the steepest drop in the loss. A learning rate of 2e − 2 was chosen based on the result obtained from the analysis.

3 RESULTS AND DISCUSSION

AMD Ryzen 7-3700X CPU, Nvidia GeForce RTX 3060, VRAM 12 Gig, RAM: 64 GB, NVMe 2 SSDs constitute the experimental environment.. The programming language used to create Tensorflow models in Keras is Python 3.7. These metrics show how well the model is performing. Metrics used include:

i. Accuracy: Evaluate overall, how often the model is correctly classified. It can be described in equation 7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

The accuracy can also be calculated as $1 - e$, where e is the error rate. The accuracy of the model is found to be 99.97%.

ii. Recall: This rate value shows how often the model predicts the right value of a particular class. It is computed using equation 8:

$$Recall = \frac{TP}{FN + TP} \quad (8)$$

iii. Precision: This shows whether when the model predicts the right value. It is computed using equation 9:

$$Precision = \frac{TP}{FP + TP} \quad (9)$$

iv. F1 Score: This shows the weighted mean of precision and recall. It can be seen in equation 10.

$$F1\ Score = \frac{2\ x\ Recall x\ \ Precision}{Recall + Precision} \quad (10)$$

Table 1. Model Architecture

Layer Name	Output Size	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
conv1	112 x 112	7×7, 64, *stride* 2				
		3×3 *max pool*, *stride* 2				
conv2_x	56 × 56	$\begin{bmatrix} 3 & x & 3,64 \\ 3 & x & 3,64 \end{bmatrix} x\,2$	$\begin{bmatrix} 3 & x & 3,64 \\ 3 & x & 3,64 \end{bmatrix} x\,3$	$\begin{bmatrix} 3 & x & 3,64 \\ 3 & x & 3,64 \end{bmatrix} x\,3$	$\begin{bmatrix} 3 & x & 3,64 \\ 3 & x & 3,64 \end{bmatrix} x\,3$	$\begin{bmatrix} 3 & x & 3,64 \\ 3 & x & 3,64 \end{bmatrix} x\,3$
conv3_x	28 × 28	$\begin{bmatrix} 3 & x & 3,128 \\ 3 & x & 3,128 \end{bmatrix} x\,2$	$\begin{bmatrix} 3 & x & 3,128 \\ 3 & x & 3,128 \end{bmatrix} x\,4$	$\begin{bmatrix} 3 & x & 3,128 \\ 3 & x & 3,128 \end{bmatrix} x\,4$	$\begin{bmatrix} 3 & x & 3,128 \\ 3 & x & 3,128 \end{bmatrix} x\,4$	$\begin{bmatrix} 3 & x & 3,128 \\ 3 & x & 3,128 \end{bmatrix} x\,8$
conv4_x	14 × 14	$\begin{bmatrix} 3 & x & 3,256 \\ 3 & x & 3,256 \end{bmatrix} x\,2$	$\begin{bmatrix} 3 & x & 3,256 \\ 3 & x & 3,256 \end{bmatrix} x\,6$	$\begin{bmatrix} 3 & x & 3,256 \\ 3 & x & 3,256 \end{bmatrix} x\,6$	$\begin{bmatrix} 3 & x & 3,256 \\ 3 & x & 3,256 \end{bmatrix} x\,23$	$\begin{bmatrix} 3 & x & 3,256 \\ 3 & x & 3,256 \end{bmatrix} x\,36$
conv5_x	7 × 7	$\begin{bmatrix} 3 & x & 3,512 \\ 3 & x & 3,512 \end{bmatrix} x\,2$	$\begin{bmatrix} 3 & x & 3,512 \\ 3 & x & 3,512 \end{bmatrix} x\,3$	$\begin{bmatrix} 3 & x & 3,512 \\ 3 & x & 3,512 \end{bmatrix} x\,3$	$\begin{bmatrix} 3 & x & 3,512 \\ 3 & x & 3,512 \end{bmatrix} x\,3$	$\begin{bmatrix} 3 & x & 3,512 \\ 3 & x & 3,512 \end{bmatrix} x\,3$
Classification Layer	1 × 1	*average pool*, 29D *fully connected*, *softmax*				
FLOPS		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

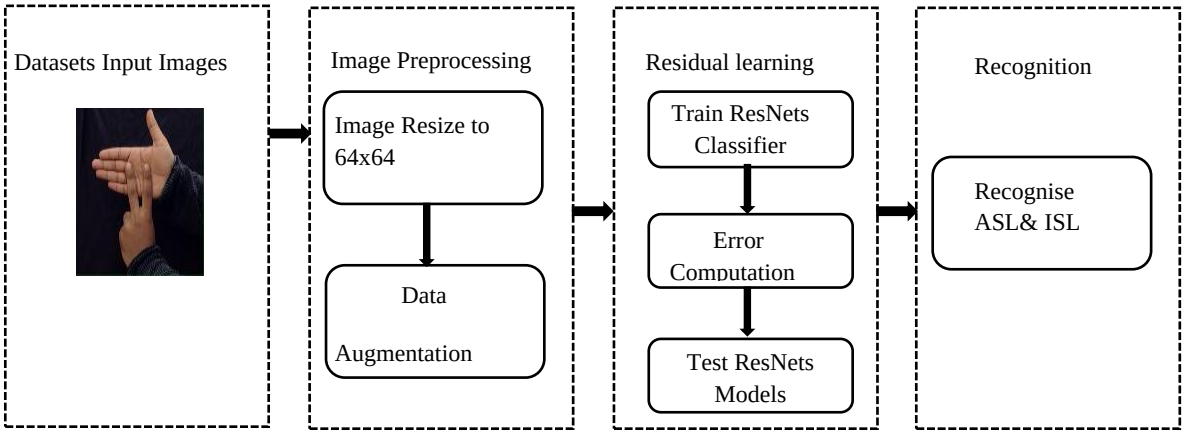

Fig 3. Block diagram of Sign Language Recognition System

The training and testing accuracy result of different ResNet Models on the ASL Dataset is shown in Table 2 and that of the ISL dataset is shown in Table 3. Figs 4 to 5 observed the ResNet models' performances on ASL and ISL datasets.

Table 2 Comparison Results of different ResNet Models on ASL Dataset

Models	Training Accuracy	Testing Accuracy	F1 Score
ResNet18	0.9877	0.9	0.9438
ResNet34	0.9901	0.91	0.950
ResNet50	0.9911	0.99	0.990
ResNet101	0.992	0.99	0.991
ResNet152	0.9968	0.99	0.9934

Table 3. Comparison Results of different ResNet Models on ISL Dataset

Models	Training Accuracy	Testing Accuracy	F1 Score
ResNet18	0.93	0.92	0.925
ResNet34	0.941	0.9027	0.9218
ResNet50	0.9662	0.9195	0.9428
ResNet101	0.9594	0.9354	0.9474
ResNet152	0.9798	0.9736	0.9767

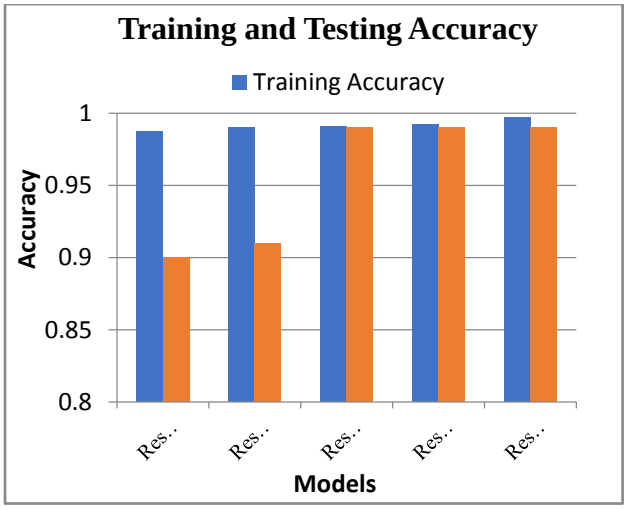

Fig 4. Accuracy results of ResNet models on ASL Dataset

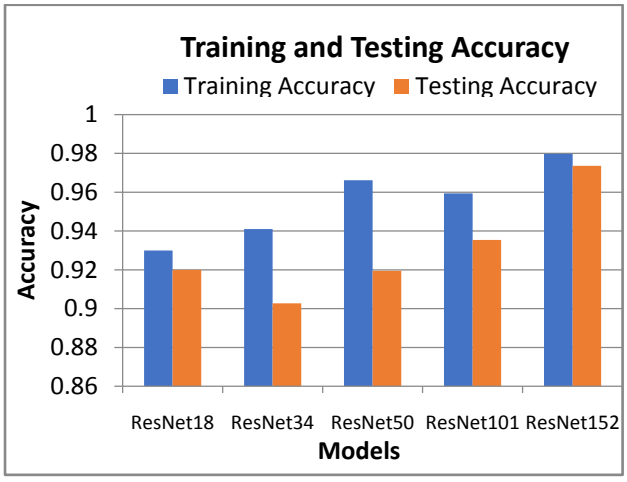

Fig. 5. Accuracy results of ResNet models on ISL Dataset

From table 2, the analysis revealed that the ResNet models had great performance on the ASL dataset as against the ISL dataset. Specifically the ResNet models; Resnet18, ResNet34, ResNet50, ResNet101 and ResNet152 recorded 0.93, 0.941, 0.9662, 0.9594 and 0.9798 as training accuracy values, respectively. From the testing, maximum accuracies of 0.9, 0.91, 0.99, 0.99 and 0.99 were attained for the Resnet18, ResNet34, ResNet50, ResNet101 and ResNet152 respectively. It demonstrates that the ResNet architectures with much more depth (ResNet50 and ResNet101) perform well, while ResNet152 obtains the overall performance. The interpretation here is that the ResNet152 which performed best had the lowest misclassification error of 0.0202 and 0.01 for training and testing respectively. This shows that when ResNet152 is applied to the ASL dataset, an expected image classification accuracy of almost 100% can be achieved

The capability of variants of ResNet models was further evaluated on a complex ISL dataset. In general, the ResNet models suffer slightly in terms of prediction accuracy on ISL data. ResNet18, ResNet34, ResNet50 and ResNet101 produced accuracy values of 0.93, 0.941, 0.9662 and 0.9594 respectively, as shown in Table 2. Again, ResNet152 yielded the best performance with 0.9798 and 0.9736 as training and test accuracy values, respectively. The accuracy values imply misclassification errors of 0.0032 and 0.01 for training and testing. The nature of the dataset accounts for the variation in the performance of the ResNet Models.

We also compute the F1 score using the test data, as seen in Fig. 4. Across all ResNet designs, very good F1 scores were obtained on the ASL dataset when compared to the ISL dataset seen in Table 5. For these two datasets, the performance of ResNet152 dominates other ResNet variants, obtaining 99.34% and 97.36% for the ASL and ISL datasets, respectively. This affirmed assertion by He et al. [16] that ResNets can easily gain accuracy through high depth, resulting in better results. Also, the model network parameters were modified for the various ResNet models and it was observed that they are all the more improved in terms of recognition capabilities for double-hand images with complex shapes. Generally, the models were able to classify images into the respective classes accurately with minimal false predictions.

4. CONCLUSION

This paper presents a comparative study among five different variants of residual models to solve the problem of sign language recognition. The performance of the ResNet models (Resnet18, ResNet34, ResNet50, ResNet101 and ResNet152) are tested on two public datasets. The first dataset mainly ASL was used to test the single-hand recognition capabilities whereas the second dataset mainly, ISL was used to test the ability to recognise double-hand sign language. Experimental results show the 152-layer ResNet model achieved the topmost accuracy for the different datasets. This suggests ResNet152 is more capable of handling single and double-handed datasets than Resnet18, ResNet34, ResNet50, and ResNet101. Thus, the ResNet152 is a good predictor and therefore highly recommended for images with complex shapes.

It is recommended that future works should extend to identifying real-time data of hand signs hence, capturing signs that require dynamic movements of the hands.

5. REFERENCES

[1] Bickenbach, J.E., Cieza, A. and Sabariego, C., (2016), "Disability and Public Health" Int. J. Environ. Res. Public Health, Vol. 13, pp. 123-132.

[2] Groce, N.E., 2018. Global disability: an emerging issue. The Lancet Global Health, 6(7), pp.e724-e725.

[3] Agangiba, M., "Accessibility of E-government Services for Persons with Disabilities in Developing Countries-The Case of Ghana ", Unpublished Doctoral Thesis, Department of Information Systems, University of Cape Town, South Africa, 290pp.

[4] Gedam, S. and Shrawankar, U. (2017), "Challenges and opportunities in fingerspelling recognition in the air", In International Conference on Innovative Mechanisms for Industry Applications, Bengaluru, India, pp. 60 – 65.

[5] Nair, A.V. and Bindu, V., (2013), "A review on Indian sign language recognition", International journal of computer applications, Vol. 73, No. 22, pp. 33-38.

[6] Mahesh, M., Jayaprakash, A. and Geetha, M., 2017, September. Sign language translator for mobile platforms. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1176-1181). IEEE.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[8] Bousbai, K. and Merah, M., (2019), "A Comparative

Study of Hand Gestures Recognition Based on MobileNetV2 and ConvNet Models", In International Conference on Image and Signal Processing and their Applications (ISPA), Mostaganem, Algeria, pp. 1-6.

[9] Kusters, A., De Meulder, M., and O'Brien, D. (2017), Innovations in deaf studies: The role of deaf scholars, Oxford University Press, 416 pp.

[10] Bhujbal, V.P., and Warhade, K.K., (2018), "Hand sign recognition-based communication system for speech disable people", ICICCS 2018, In Proceedings of the 2nd International Conference on Intelligent Computing and Control Systems, Madurai, India, pp. 348 – 352.

[11] Singleton, J. L., Remillard, E. T., Mitzner, T. L., and Rogers, W. A. (2019), "Everyday technology use among older deaf adults" Disability and Rehabilitation: Assistive Technology, Vol. 14, No. 4, pp. 325-332.

[12] Dhiman, R., Joshi, G. and Krishna, C.R., (2021), "A deep learning approach for Indian sign language gestures classification with different backgrounds" In Journal of Physics: Conference Series, Vol. 1950, No. 1, pp. 1-15

[13] Mannan, A., Abbasi, A., Javed, A. R., Ahsan, A., Gadekallu, T. R., & Xin, Q. (2022). Hypertuned deep convolutional neural network for sign language recognition. Computational Intelligence and Neuroscience.

[14] Lum, K.Y., Goh, Y.H. and Lee, Y.B., 2020. American Sign Language recognition based on MobileNetV2. Adv. Sci. Technol. Eng. Syst., 5(6), pp.481-488.

[15] Agrawal, M., Ainapure, R., Agrawal, S., Bhosale, S., & Desai, S. (2020, October). Models for hand gesture recognition using deep learning. In 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA) (pp. 589-594). IEEE.

[16] He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", In Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 770 – 778.

[17] Rathi, P., Kuwar Gupta, R., Agarwal, S. and Shukla, A., 2020, February. Sign language recognition using resnet50 deep neural network architecture. In 5th International Conference on Next Generation Computing Technologies (NGCT-2019).

[18] Saleh, Y. & Issa, G. (2020). "Arabic Sign Language Recognition through Deep Neural Networks Fine-Tuning". International Association of Online Engineering. https://www.learntechlib.org/p/217934/. Accessed: 21 June 2022

[19] Alleema, N., & Chandrasekaran, S. (2022). Recognition of American Sign Language Using Modified Deep Residual CNN with Modified Canny Edge Segmentation.

[20] Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q. (2016), "Deep networks with stochastic depth", In Conference on Computer Vision, Amsterdam, The Netherlands, pp. 646–661.

[21] Veit, A.; Wilber, M.J.; Belongie, S. (2016), "Residual networks behave like ensembles of relatively shallow networks", In Advances in Neural Information Processing Systems; NIPS, Montreal, QC, Canada, pp. 550–558.

[22] Wu, Z., Shen, C., Van Den Hengel, A. (2019), "Wider or deeper: Revisiting the resnet model for visual recognition", Pattern Recognition, Vol. 90, 119-133.

[23] Glorot, X.; Bordes, A.; Bengio, Y, (2011), "Deep sparse rectifier neural networks", In International Conference on Artificial Intelligence and Statistics, Lauderdale, FL, USA, pp. 315–323.

[24] Ioffe, S., Szegedy, C. (2015), "Batch normalization: Accelerating deep network training by reducing internal covariate shift", www.arxiv.org. Accessed: September 15, 2021.

[25] Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. (2017), "Aggregated residual transformations for deep neural networks", In Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp. 1492–1500.

[26] Akash, K., (2016), "Image data set for alphabets in the American Sign Language", www.kaggle.com. Accessed: August 4, 2021.

[27] Sonawane V. (2018), Indian Sign Language Dataset. www.kaggle.com. Accessed: August 20, 2021.

[28] Khun, M. and Johnson, K. (2013), Applied Predictive Modeling, Springer, Basel, 600pp.

[29] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), "ImageNet Classification with Deep Convolutional Neural Networks", In Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA, pp. 1097 – 1105.

[30] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986), "Learning representations by back-propagating errors", Nature, Vol. 323, pp. 533 – 536.

[31] Loshchilov, I. and Hutter, F. (2019), "Decoupled Weight Decay Regularization", In International Conference on Learning Representations (ICLR), New Orleans, Louisiana, USA, pp. 1-19.

[32] Kingma, D. P. and Ba, J. (2015), "Adam: A Method for Stochastic Optimization" In International Conference on Learning Representations (ICLR), San Diego, CA, USA, pp. 1-15.

[33] Wilson, A. C., Roelofs, R., Stern, M., Srebro, N. and Recht, B. (2017), "The Marginal Value of Adaptive Gradient Methods in Machine Learning", In Conference on Neural Information Processing System, Long Beach, CA, USA, pp. 1-14.

[34] Smith, L. N. (2018), A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay, US Naval, 21pp.