Sign Language Recognition Using Convolutional Neural Networks

Jarosław Kobiela
Gdańsk University of Technology, Department of Software Engineering
Gdańsk, Poland *jarokobiela@gmail.com*

Dariusz Kobiela
Gdańsk University of Technology, Department of Software Engineering
Gdańsk, Poland *dariusz.kobiela@eti.pg.edu.pl*

Adam Artemiuk
Gdańsk University of Technology
Gdańsk, Poland *s165196@student.pg.edu.pl*

Abstract

The objective of this work was to provide an app that can automatically recognize hand gestures from the American Sign Language (ASL) on mobile devices. The app employs a model based on Convolutional Neural Network (CNN) for gesture classification. Various CNN architectures and optimization strategies suitable for devices with limited resources were examined. InceptionV3 and VGG-19 models exhibited negligibly higher accuracy than our own model, but they also had more complicated architectures. The best method for network optimization became Layer Decomposition which achieved the lowest inference time in classification effectiveness. Each optimization method reduced the inference time of our model at the small expense of classification accuracy. The accelerators with the shortest inference time were GPU and CPU in a configuration of 5 threads. For the purpose of loading the trained models, running and testing their effectiveness under different hardware configurations a prototype of the mobile application was developed.

Keywords: Sign language, Convolutional Neural Network (CNN), Quantization Aware Training (QAT), Layer Decomposition, Knowledge Distillation.

1. Introduction

With nearly 5% of the global population experiencing hearing loss [14], the need for effective communication tools for the deaf and hard-of-hearing community is paramount. Sign language serves as a vital means of communication, and the automatic recognition of hand gestures can bridge the gap between sign language users and those who do not understand it.

Recent advancements in deep learning, particularly with CNNs, have significantly improved the accuracy and robustness of automatic sign language recognition systems. However, deploying such systems on mobile devices presents unique challenges due to the limited computational resources and power constraints of these platforms [3].

This research focuses on the development of a mobile application that recognizes static hand gestures from the American Sign Language (ASL). We explore various CNN architectures and optimization techniques, such as Post-Training Quantization (PTQ) [15], Quantization Aware Training (QAT) [6], Layer Decomposition [9], and Knowledge Distillation [11], to create a model that is both accurate and efficient. By prioritizing static gestures, we simplify the recognition task and reduce the computational complexity required, making it more feasible for deployment on mobile devices. The provided application has the potential to improve social interactions and empower individuals who rely on sign language for communication.

2. Background and related work

Among the models designed for gesture recognition, we can distinguish modified Google InceptionV3 (used by Malakan & Albaqami [10] in order to distinguish 26 gestures corresponding to letters between A and Z), modified Inception V3 introduced by Hasan et al. [5] (with the aim of detecting the alphabet and numbers in ASL), modified VGG-19 model (applied by Le et al. [8] for recognizing digits from 0 to 9), and model implemented by Bhadra & Kar [2] consisting of Deep Multi-Layer CNN for detection of static and dynamic gestures representing descriptions of actions, emotions, and positions. Another approach of gesture recognition was shown in Ionescu et al. [7], where authors proposed novel approach for inspired by a distance measure for strings called Local Rank Distance (LRD). Rahim et al. [12] developed a model for real-time gesture recognition using CNN and an SVM (Support Vector Machine) classifier. The dataset consisted of 20 isolated gestures with dimensions of 200x200, totaling 18,000 samples. Each image underwent segmentation using the YCbCr color space and SkinMask segmentation, which involved converting the color space to HSV, segmenting the skin color, and performing morphological image processing. Two convolutional networks were used for feature extraction. The input to each network was the images segmented by YCbCr and SkinMask, respectively. The convolutional layers and network parameters were identical but differed in weights. The network consisted of two pairs of convolutional and pooling layers and a flattening layer. Next model, presented by Sun et al. [13], performed real-time gesture detection of 10 digits. The dataset consisted of approximately 16,000 images. First, to locate the hand in the image, segmentation was performed. The proposed method used an AdaBoost classifier based on Haar features. In the next step, the CamShift algorithm was used to track hand gestures in real-time. The LeNet-5 architecture was used for gesture classification. The detailed report in the GitHub repository includes a comprehensive table of models that implement gesture recognition tasks, as well as a table listing the most commonly used datasets for training these models.

3. Research method

The focus was placed on the static gestures due to the lower complexity of the model required for recognizing such gestures which translates to easier implementation and further optimization.

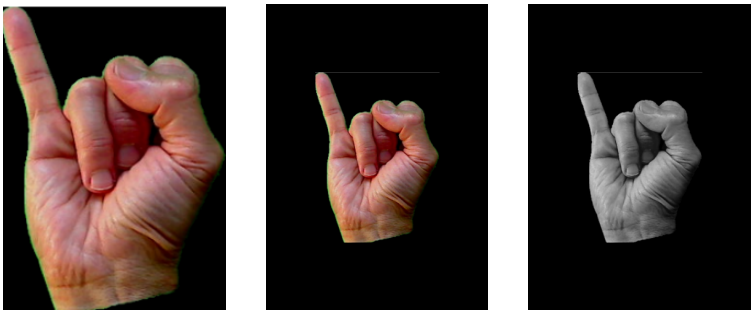
3.1. Dataset, data augmentation and processing

It was decided to use Massey dataset [1] developed by Bartczak et al., which contains 2524 samples of static gesture photos typical for ASL. The dataset contains 36 different gestures representing the alphabet letters and numbers. All images are in color, in PNG format, and feature a separated hand displaying the respective gesture on a black background (see Fig. 1a). Due to the small number of data samples, the Massey dataset underwent augmentation, including modifications such as image brightness adjustment, shifting of the hand in the image, and zooming in or out of the hand in the image (see Fig. 1b). Further data preprocessing steps involved converting the color space from RGB to HSV in order to separate the hand from the background, converting the images from PNG to JPG format and processing into grayscale (see Fig. 1c). These operations were aimed at accelerating the training of the model.

3.2. Model

According to the findings of the literature review[1], for further experiments, a model with the highest accuracy value (99.2%, as introduced by Hasan et al. [4] using a dataset consisting of 36

[1] A detailed report on the literature review and all the code are available at our GitHub repository: https://github.com/DariuszKobiela/sign-language-recognition-using-CNN



(a) original dataset **(b)** after augmentation **(c)** after final processing

Fig. 1. Sample data from the processed dataset

letters of the Bengali language) was used and modified, finally consisting of 15 convolutional layers and multiple other types of layers. Results obtained by the chosen and modified model were compared with the baseline models: modified InceptionV3 [5], modified VGG-19 [8], and model developed by Bhadra & Kar [2].

3.3. Mobile Application Testing the Network

Once the neural network models have been trained, they were tested using a mobile app developed specifically for this research. The main goal of the app was to allow loading of the trained models, running and testing their effectiveness under different hardware configurations.

3.4. Methods of Network Optimization

Network optimization involved applying various techniques to reduce the computational complexity of the CNN model without significantly compromising its accuracy. Four main optimization methods were explored in this research:

- Post-Training Quantization (PTQ) involves converting the model's weights and activations from higher precision to lower precision after the model has been trained [15];
- Quantization Aware Training (QAT) incorporates quantization into the training process [6];
- Layer Decomposition optimizes each layer's compression ratio in a deep neural network to meet overall compression goals, slicing channels into groups and applying low-rank decomposition [9];
- Knowledge Distillation involves training a smaller, more efficient "student" model to mimic the behavior of a larger, more complex "teacher" model [11].

3.5. Evaluation metrics

We evaluated the trained models based on inference time [ms] and gesture prediction accuracy. The study consisted of approximately 5-seconds long trials of recognizing each gesture, using various combinations of delegates and number of threads (for CPU as the delegate).

4. Results

The following results in Tables 1, 2 and 3 show the performance of investigated models.

Table 1. Comparison of test accuracy between our own and baseline models

Model	InceptionV3 [5]	VGG-19 [8]	Bhadra & Kar model [2]	Own model
Test Accuracy[%]	98.60	98.28	95.73	98.13

Table 2. Performance evaluation of our model with various optimization techniques

Model	Accuracy test [%]	Parameters	The median inference time CPU-5	The median inference time GPU
Own model	98.13	37,703,397	431.5	277
PTQ	98.13	37,703,397	167	290
QAT	97.17	37,716,455	154	298
Layer decomposition	97.80	7,995,717	60	55
Knowledge distillation	97.96	4,100,709	86	61

Table 3. Median inference time for our own model optimized using layer decomposition method

accelerator + number of threads (for CPU)	CPU-1	CPU-5	CPU-10	GPU
median inference time [ms]	134	60	149	55

5. Discussion and conclusions

As InceptionV3 and VGG-19 models exhibited negligibly higher accuracy (see Table 1), but they also had more complicated architectures, we decided to further streamline our own model in the subsequent stages. The Bhadra & Kar model [2] was discarded because of its poor performance. In case of real-world samples our model at times struggled to recognize a gesture shown in the same way as for the baseline models. Each optimization method reduced the inference time of our baseline model at the small expense of classification accuracy. In real-time testing scenarios, the optimized models managed to recognize most of the types of static gesture. Layer decomposition emerged as the most effective optimization method, achieving the lowest inference times while maintaining strong classification performance. Some static gestures from the ASL alphabet are very similar to each other and are distinguished by small details (such as a different thumb position), which makes their correct classification difficult. Examples of such gestures include '0' and 'O', '2' and 'V', or 'K', '6' and 'W'. As the distance between the hand and the camera increases, the effectiveness of gesture recognition becomes increasingly challenging, resulting in a higher number of false results. This may happen due to the small dimensions of the input image, causing the hand to appear smaller, as well as the phenomenon of slight blurring of the image resulting from hand segmentation. These two factors can make the distinguishing details of a particular gesture practically invisible.

The introduction of hand segmentation in the preprocessing stage significantly improved the accuracy of gesture inference. By isolating the hand from the background, the influence of environmental factors on prediction results was mitigated, and real-time samples more closely resembled the training dataset. Segmentation using the HSV color conversion method proved effective in isolating the hand. However, this method's effectiveness is limited when the background contains elements with H, S, and V values within the selected ranges. In such cases, these elements in addition to the hand itself also appear in the resulting image increasing the risk of false results. During the conducted research it was observed that in some cases elements such as bright walls and other body parts (e.g., face) decreased detection accuracy because these

elements appeared in the resulting image and caused interference. It is also worth considering that the human hand can have various colors and much depends on the current ambient lighting, which can cause the hand in the image to appear brighter or darker. There is therefore a risk that certain colors will not fall within the defined thresholds resulting in the loss of information that may be crucial during the inference process.

References

[1] Barczak, A. L. C., Reyes, N. H., Abastillas, M. E., Piccio, A., and Susnjak, T.: A New 2D Static Hand Gesture Colour Image Dataset for ASL Gestures. In: 2011. URL: https://api.semanticscholar.org/CorpusID:8908937.

[2] Bhadra, R. and Kar, S.: Sign Language Detection from Hand Gesture Images using Deep Multi-layered Convolution Neural Network. In: *2021 IEEE Second International Conference on Control, Measurement and Instrumentation (CMI)*. 2021, pp. 196–200.

[3] Glegoła, W., Karpus, A., and Przybyłek, A.: MobileNet family tailored for Raspberry Pi. In: *Procedia Computer Science* 192 (2021), pp. 2249–2258.

[4] Hasan, M. M., Srizon, A. Y., and Hasan, M. A. M.: Classification of Bengali Sign Language Characters by Applying a Novel Deep Convolutional Neural Network. In: *2020 IEEE Region 10 Symposium (TENSYMP)*. 2020, pp. 1303–1306.

[5] Hasan, M. M., Srizon, A. Y., Sayeed, A., and Hasan, M. A. M.: Classification of American Sign Language by Applying a Transfer Learned Deep Convolutional Neural Network. In: *23rd International Conference on Computer and Information Technology*. 2020, pp. 1–6.

[6] Huang, X., Liu, Z., Liu, S.-Y., and Cheng, K.-T.: *Efficient Quantization-aware Training with Adaptive Coreset Selection*. 2023. arXiv: 2306.07215 [cs.LG].

[7] Ionescu, R. T., Popescu, M., Conly, C., and Athitsos, V.: Local frame match distance: A novel approach for exemplar gesture recognition. In: Aug. 2017, pp. 788–792.

[8] Le, S., Lei, Q., Wei, X., Zhong, J., Wang, Y., Zhou, J., and Wang, W.: Smart Elevator Cotrol System Based on Human Hand Gesture Recognition. In: *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*. 2020, pp. 1378–1385.

[9] Liebenwein, L., Maalouf, A., Feldman, D., and Rus, D.: Compressing Neural Networks: Towards Determining the Optimal Layer-wise Decomposition. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 5328–5344.

[10] Malakan, Z. M. and Albaqami, H. A.: Classify, Detect and Tell: Real-Time American Sign Language. In: *4th National Computing Colleges Conference*. 2021, pp. 1–6.

[11] Phuong, M. and Lampert, C.: Towards understanding knowledge distillation. In: *36th International conference on machine learning*. 2019, pp. 5142–5151.

[12] Rahim, M. A., Islam, M. R., and Shin, J.: Non-Touch Sign Word Recognition Based on Dynamic Hand Gesture Using Hybrid Segmentation and CNN Feature Fusion. In: *Applied Sciences* 9.18 (2019).

[13] Sun, J.-H., Ji, T.-T., Zhang, S.-B., Yang, J.-K., and Ji, G.-R.: Research on the Hand Gesture Recognition Based on Deep Learning. In: Dec. 2018, pp. 1–4.

[14] WHO: Deafness and hearing loss (02.04.2024). In: (). URL: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.

[15] Zhang, J., Zhou, Y., and Saab, R.: *Post-training Quantization for Neural Networks with Provable Guarantees*. 2023. arXiv: 2201.11113 [cs.LG].