Proceedings of the 2023 International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/40/20230636

Edge impulse-based convolutional neural network for Hand Posture Recognition

Yiwei Gui

School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

20204321371@sr.gxmu.edu.cn

Abstract. Hand Posture Recognition (HPR) plays a crucial role in enabling effective human-computer interaction, particularly for individuals with hearing disabilities. The study compares five models, including MobileNetV2 96x96 0.35, MobileNetV1 96x96 0.25, MobileNetV1 96x96 0.1, self-designed Network 1, and self-designed Network 2, based on the Sébastien Marcel Static Hand Posture Database. Evaluation metrics - infserencing time, peak RAM usage, flash usage, and accuracy - are used to analyze the performance. The experiment workflow for each model comprises five major steps. Firstly, a random selection of 120 images from the Sébastien Marcel Static Hand Posture Database is converted to JPG format. Then, the images are divided into 80% training data and 20% testing data. Subsequently, the original images are normalized, and features are extracted for further processing. Subsequently, the models are individually trained using the preprocessed data, optimizing their parameters. Finally, the trained models are evaluated using the testing data set to assess their performance in hand posture recognition. The results indicate that MobileNetV2 96x96 0.35 achieves the highest accuracy of 96.69% while consuming fewer hardware resources compared to other models. MobileNetV1 96x96 0.1 demonstrates the lowest inferencing time and peak RAM usage, making it suitable for real-time applications. Furthermore, self-designed Model 1 exhibits the lowest flash usage, making it a viable option for resource-constrained devices. This study provides valuable insights into the selection of CNN architectures for HPR, offering guidance for practitioners to choose models based on specific application requirements.

Keywords: Computer Vision, Machine Learning, Image Classification, Hand Posture Recognition.

1. Introduction

By 2023, the World Health Organization (WHO) estimates that approximately 430 million individuals, comprising 432 million adults and 34 million children, will be in need of rehabilitation to address their disabling hearing loss. This accounts for over 5% of the global population, highlighting the significant impact and prevalence of this condition worldwide [1]. Sign language connects people having hearing disabilities with family members, educators, interpreters, and support personnel associated with the deaf community. However, it comes with certain drawbacks, including limited spread, steep learning curve, inapplicable to hearing individuals, grammar and expression limitations, and cultural barriers. Hand Posture Recognition (HPR) is of paramount importance across various domains, including human-computer interaction, virtual reality, gaming, education, accessibility, and medical

DOI: 10.54254/2755-2721/40/20230636

rehabilitation. By accurately detecting and interpreting hand gestures, machines can facilitate seamless and natural communication between individuals and computers, thereby revolutionizing the way individuals interact with technology.

In the early HPR field, there were mainly two classes of methods: feature-based methods and appearance-based methods. Feature-based methods focus on the number and location of fingers. Cao and Li [2] proposed a HPR algorithm based on topological features, which has high recognition accuracy. They chose to use topological features because they are stable and can be computed even for undefined gestures. Axak et al. [3] developed a virtual mouse system that can recognize gestures using convexity analysis of contours. However, this method only works for some example gestures. Tusor and Varkonyi-Koczy [4] use a fuzzy neural network to select feature points, but it requires a lot of time and hardware resources to achieve good results. On the other hand, appearance-based methods focus on the overall appearance of the hand. Suryanarayan et al. [5] achieved scale- and rotation-invariant gesture recognition using 3D volumetric shape descriptors. Furthermore, Malassiotis and Stintzis [6] used range data for static gesture recognition, achieving 70% to 90% accuracy on 20 gestures. However, none of the aforementioned studies address the challenge of achieving high accuracy while minimizing resource usage.

In this regard, this paper compares performance of five different Convolutional Neural Network (CNN) on the same data set of hand posture in order to find the optimal architecture, which has relatively high accuracy and takes relatively few hardware resources. Specifically, the five CNN models chosen for this article are MobileNetV2 96x96 0.35, MobileNetV1 96x96 0.25, MobileNetV1 96x96 0.1, the self-designed Network 1, and the self-designed Network 2. The dataset is Sébastien Marcel Static Hand Posture Database [7]. Through evaluating the performance of the neural network from inferencing time, peak ram usage, flash usage, and accuracy, this paper finds that MobileNetV2 96x96 0.35 is the best model with an accuracy of 96.69% and least hardware resources.

2. Method

2.1. Dataset preparation

The Sébastien Marcel Static Hand Posture Database [7] used in this study is a comprehensive collection of hand posture RGB images captured from various angles and positions. There are six different categories of hand posture in this database, and they represent ‘A’, ‘B’, ‘C’, ‘FIVE’, ‘POINT’, and ‘V’ shown in Figure 1 respectively. The total number of images in the original dataset is 4,872. In this study, 200 images are selected from each category randomly and therefore the total number of images in the dataset this study uses are 1200. Originally, the images are all based on ppm format. In this article, all the images are converted into JPG format in order to fulfil the requirement of Edge Impulse platform. Individual images in the original dataset vary in size, from 50×50 to 100×100. In this study, the size of all images is limited to 96×96 to ensure the convenience of normalizing. Moreover, during the normalization process, the value of each pixel’s channel in the image is transformed into a floating-point number ranging from 0 to 1. If the image is in grayscale, each pixel is converted into a single value using the ITU-R BT.601 conversion, which considers only the luminance (Y’) component.

Figure 1. Examples of six hand postures representing ‘A’, ‘B’, ‘C’, ‘FIVE’, ‘POINT’, and ‘V’ from left to right.

DOI: 10.54254/2755-2721/40/20230636

2.2. Edge impulse-based MobileNet for HPR

2.2.1. Introduction to Edge Impulse. Edge Impulse is an integrated development platform that facilitates the creation, training, and deployment of machine learning models for edge devices. The term "edge" refers to computing that occurs closer to the data source or device, rather than in the cloud. This approach has gained traction due to its ability to process data locally, reducing latency, preserving privacy, and conserving bandwidth.

2.2.2. Introduction to MobileNet. MobileNet is a collection of neural network architectures that are specifically engineered for mobile and resource-limited devices, with an emphasis on being lightweight and efficient. These architectures prioritize efficiency and low computational demands while maintaining a reasonable level of accuracy in tasks such as image classification, image recognition [8]. MobileNetV2 96x96 0.35 [9,10] is a variant in the MobileNetV2 family designed specifically for image inputs with 96x96 pixel resolution with a width multiplier of 0.35. This means that the model is suitable for processing smaller resolution images while maintaining efficiency and lightness, e.g., for real-time face recognition or gesture recognition and other resource tasks on constrained devices. MobileNetV1 is the first version of the MobileNet family and a lightweight convolutional neural network architecture for computationally resource-constrained environments. MobileNetV1 96x96 0.25 is a variant in the MobileNetV1 family designed for processing images with 96x96 pixel resolution with a 0.25 width multiplier. MobileNetV1 96x96 0.1 is also a special variant in the MobileNetV1 family optimised for image inputs with 96x96 pixel resolution and a width multiplier of 0.1. Although it makes a large reduction in model capacity, it can still be used for a number of basic image analysis tasks, providing scenarios where computational resources are scarce with a lightweight visual processing solution. Apart from pre-trained model MobileNetV1 and MobileNetV2, two self-designed neural networks are implemented in this study. For the self-designed model 1, it sequentially incorporates a 2D convolutional layer with 32 filters and a kernel size of 3x3, followed by pooling, further enhancing the feature hierarchy through a subsequent 2D convolutional layer with 16 filters. The dropout rate is 0.25 in the dropout layer, which serves as a regularization mechanism. Finally, the architecture culminates in an output layer comprising 6 neurons, each representing a distinct class. For the self-designed model 2, it begins with an input layer accommodating 27,648 features, which is succeeded by two consecutive layers of 2D convolution and pooling. Each of these layers employs 32 filters with a kernel size of 3 and undergoes convolution and pooling operations. Subsequently, the feature maps traverse two more analogous 2D convolution and pooling layers. Following this, a flatten layer transforms the output into a one-dimensional vector. To mitigate overfitting, a dropout layer with a rate of 0.25 is introduced, providing regularization. The model culminates in an output layer, which represents the final classification predictions.

2.2.3. The procedure of training model. The workflow for each model is divided into five major parts. In the Data Format Converting, 120 images are randomly selected from the Sébastien Marcel Static Hand Posture Database and converted from ppm format to JPG format. In the Data Input, 80% of the images are training data and 20% is testing data by selecting randomly. In Data Preprocessing, after normalizing the original image, features are extracted from dataset and become the inputs for the Model Training. In the Model Training, five different models are trained individually, and the trained model is tested with testing data in the end. The workflow of experiment for each model is demonstrated in Figure 2.

DOI: 10.54254/2755-2721/40/20230636

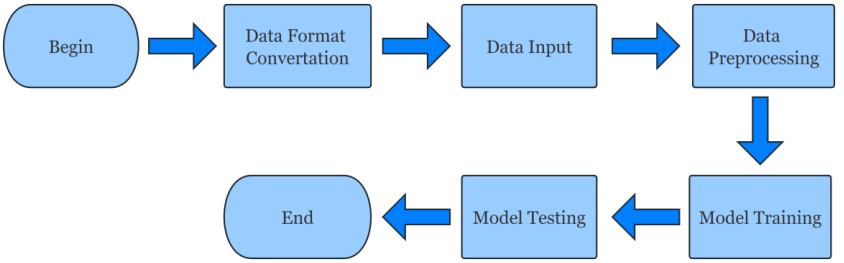

Figure 2. The workflow of experiment for each model.

2.3. Implementation details

The training of the model utilizes the Cortex-M4F 80MHz, a 32-bit embedded processor architecture developed by Arm. For each experiment in the model training, there are 20 training cycles, a learning rate of 0.0005, and a validation set size of 20%.

3. Results and discussion

The results are shown in Table 1. Inferencing time refers to the time it takes for the model to process an input and produce an output. Lower inferencing times are generally desirable, especially for real-time or time-sensitive applications; Peak RAM usage indicates the maximum amount of memory the model utilizes during inference. This metric is crucial, particularly for resource-constrained devices. Models with lower peak Random Access Memory (RAM) usage are better suited for devices with limited memory capacity; Flash usage represents the amount of storage space the model occupies in flash memory. Smaller flash usage is advantageous as it allows more models or additional data to be stored on the device; Accuracy reflects how well the model performs in terms of correctly classifying inputs.

Table 1. Performance of five models.

Model Type	Inferencing Time (ms)	Peak Ram Usage (KB)	Flash Usage (KB)	Accuracy (%)
MobileNetV2 96x96 0.35	424	947.8	1600	96.69
MobileNetV1 96x96 0.25	163	321.1	862.8	74.38
MobileNetV1 96x96 0.1	86	149.1	180.9	14.46
Self-designed Model 1	2,410	363.3	84.7	90.91
Self-designed Model 2	3,021	363.4	143.5	95.04

It is clearly that MobileNetV2 96x96 0.35 can achieve the highest accuracy of 96.69%, with 424ms inferencing time, 947.8k peak ram usage, and 1.6m flash usage. Besides, MobileNetV1 96x96 0.1 takes the least inferencing time of 86ms and peak ram usage of 149.1K. In addition, the Self-designed Model 1 takes the least flash usage of 84.7K.

Comparing MobileNetV2 96x96 0.35 with MobileNetV2 96x96 0.1, the inferencing time the former takes is 4.9 times that of the latter, but the accuracy of the former is 6.9 times that of the latter. The reason for this is that the width multiplier is a hyperparameter that scales the number of channels in each layer of the neural network. A width multiplier of 0.35, 0.25, and 0.1 means that the number of channels in each layer is reduced to 35%, 25%, and 10% of the original MobileNetV2 architecture. Certainly, though MobileNetV2 96x96 0.35 take more hardware resources, its accuracy can be as possible as close to that of the original MobileNetV2 architecture.

Comparing MobileNetV2 96x96 0.35 with Self-designed Model 1 which consist of a 2D convolutional layer with 32 filters and a 2D convolutional layer with 16 filters, the flash usage of the former is 19 times that of latter. MobileNetV2, even though simplified to 35%, is still a massive and

DOI: 10.54254/2755-2721/40/20230636

complicated neural network compared with the two-layers network. Therefore, MobileNetV2 96x96 0.35 has to take much more hardware resources.

The choice of model depends on the specific requirements of the application. MobileNetV2 96x96 0.35, despite requiring more hardware resources, offers the highest accuracy and can be suitable for applications where accuracy is critical. MobileNetV1 96x96 0.1, with its low inferencing time and peak RAM usage, is more suitable for real-time or time-sensitive applications. Self-designed Model 1, with its low flash usage, is ideal for resource-constrained devices.

To further optimize these models, future considerations could involve exploring different width multipliers, such as intermediate values between 0.35 and 0.1 for MobileNetV2, to find a better balance between accuracy and resource utilization. Additionally, exploring alternative model architectures e.g. ResNet and VGG [11, 12] that can achieve comparable accuracy with reduced hardware resource requirements would also be worth investigating.

4. Conclusion

This work aims to find the optimal architecture of CNN model for hand posture recognition. The study implements five different CNN models based on Edge Impulse platform and compares them with each other based on inferencing time, peak ram usage, flash usage, and accuracy. The selection of a model depends on the specific needs of the application. MobileNetV2 96x96 0.35 excels in accuracy, MobileNetV1 96x96 0.1 is optimal for real-time applications, and Self-designed Model 1 is advantageous for resource-constrained devices. Looking ahead, future optimization efforts could entail delving into various width multipliers, such as 0.2 and 0.25, among others. Additionally, researchers may explore alternative model architectures as a means to achieve a more optimal equilibrium between accuracy and resource utilization.

References

[1] World Health Organization 2023 Deafness and hearing loss https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss.
[2] Zhang Z 2018 Research on vision-based real-time dynamic gesture segmentation method (in Chinese) master thesis Henan University.
[3] Wan K 2013 Research and Application of Gesture Recognition System (in Chinese) master thesis Guangdong University of Technology.
[4] Qi B 2011 Research on gesture recognition algorithm based on dynamic fuzzy neural network (in Chinese) master thesis Southwest University.
[5] Xu X 2018 Research on the method of dynamic gesture recognition based on 3D deep neural network (in Chinese) master thesis Xidian University.
[6] Malassiotis S and Strintzis M G 2008 Real-time hand posture recognition using range data *Image and Vision Computing* p 1027-1037.
[7] Marcel S 1999 Hand posture recognition in a body-face centered space *CHI'99 Extended Abstracts on Human Factors in Computing Systems* p 302-303.
[8] Howard A G et al 2017 Mobilenets: Efficient convolutional neural networks for mobile vision applications *arXiv preprint arXiv:1704.04861*.
[9] Sandler M et al 2018 Mobilenetv2: Inverted residuals and linear bottlenecks *In Proceedings of* the IEEE conference on computer vision and pattern recognition pp 4510-4520.
[10] Srinivasu P N et al 2021 Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM Sensors 21(8) p 2852.
[11] Qiu Y Wang J Jin Z et al 2022 Pose-guided matching based on deep learning for assessing quality of action on rehabilitation training Biomedical Signal Processing and Control 72: 103323.
[12] Nair R R Singh T Basavapattana A Pawar M M 2022 Multi-layer, multi-modal medical image intelligent fusion Multimedia Tools and Applications 81(29) 42821-42847.