A Comparative Analysis of EfficientNet and MobileNet Models' Performance on Limited Datasets: An Example of American Sign Language Alphabet Detection

Hongwen Pu[1 *] and Keyi Yi[2]

[1]Dulwich College (Singapore), Singapore, Singapore

[2]Dulwich International High School (Suzhou), Suzhou, China

*Corresponding author: hongwen.pu25@stu.dulwich.org

Abstract. This paper explored the performance of EfficientNet architecture and MobileNet architecture while processing sign language alphabets using small-scale datasets. EfficientNet and MobileNet are the two most popular architectures in the computer vision industry. The outbreak of the COVID-19 pandemic demonstrates the importance of investigating two model's performance while a product needs to be developed in a short time. Previously, research has been conducted on two models, mainly focusing on the two model's performance while handling small datasets related to medicine. However, there remains a research gap for sign language. A combined dataset obtained from two datasets from Kaggle was used to train the model. The models' performance under 5 epochs, 10 epochs, and 20 epochs were deduced and compared. In general, the performance of MobileNetV2 models is outstanding, especially under 5 epochs, while other MobileNet and EfficientNet models show an intense overfit. Moving forward, the models could be tested on more powerful platforms and more models could be compared.

Keywords: EfficientNet, MobileNet, architecture, sign language, small-scale datasets, combined datasets.

1. Introduction

EfficientNet architecture is a modern computer vision architecture based on Convolutional Neural Networks (CNN) that is well-known for its efficiency and high accuracy. It achieved a mean Top-1 Accuracy of 77.1% and a mean Top-4 Accuracy of 93.3% for its base model, EfficientNet-B0 on ImageNet [1]. MobileNet is also a modern computer vision model based on Convolutional Neural Networks (CNN), known for its exemplary performance on mobile and embedded vision applications and efficiency. Its full convolution model achieved an ImageNet accuracy of 71.7% [2].

In the field of machine learning, the dataset is paramount to train decent models. However, the data collected may be limited due to various factors. One contributing factor is the time constraints. In 2019, the sudden outbreak of COVID-19 necessitated the expeditious product development and deployment. Under this circumstance, the data collected may be limited to produce a decent computer vision model. Another contributing factor is data availability since some data may not be available or the dataset may not be diverse. For instance, it is not possible to collect historical data primarily. Data augmentation could be employed as a technique. However, data augmentation may face challenges if there is an extreme lack of original data, when the data is too complex, or when the data is too sensitive to minor changes. For example, data augmentation may not be employed while handling housing prices, as their nature is continuous, numerical values that reflect real-world economic data.

The drawbacks brought by small-scale datasets are also obvious. A small-scale dataset has a high chance of causing overfit (learning extensive patterns of the training data which causes the model to perform less well on the test data). It could also cause issues during the production environment as the model may not perform well since the dataset is not diverse.

American Sign Language is a sign language mainly used in the United States and most of Anglophone Canada, while also significantly used in Africa and South-East Asia. Like other sign languages such as Chinese Sign Language (CSL), American Sign Language uses hand gestures and facial expressions to express a word.

Most of the research that has been conducted is focused on medical scenarios based on small datasets, such as the detection of breast cancer, strawberry plant disease classification, and COVID-19 detections [3-5]. Research that compares EfficientNet and MobileNet in the context of sign languages is rare. However, the dataset of sign language may be difficult and exhausting to collect as the nature of sign language is a proper language used for communication. The model we will train in this paper is a scaled-down version of the common large model used to detect sign language as the model we are going to train is only capable of detecting alphabets.

The main purpose of this investigation is to test and compare the performance of the EfficientNet architecture and the MobileNet architecture under the constraints of time and when quick development is necessary. Another purpose of this investigation is to test the two models' performance while handling small datasets.

2. Methodology

The investigation of this essay focuses on the classification of the alphabet gestures of ASL. Data augmentation could not be performed on pictures of sign language as some of the gestures are similar to each other. Some data augmentation techniques such as rotating the pictures may cause the representation of a letter to undergo alterations. As shown in the example provided in Fig 1, the gesture on the left expresses the letter "C" while the gesture on the right expresses space.

Fig. 1 An example of two similar ASL gestures that represent different alphabets (Photo/Picture credit: Original).

2.1. Data Collection

2.1.1 Data sources

To allow both models to perform the best they could, two diverse datasets with differences such as the hand used to make gestures and background were used.

Both datasets are taken from Kaggle, a data science platform comprising rich resources including datasets, notebooks, and models [6].

One of the datasets used to train the model (Dataset A) is "American Sign Language" by Kapil Londhe [6]. The dataset consists of 10,000 RGB samples for each gesture. Although it was last updated 3 years ago, it could still be used now as the gestures for American Sign Language have not changed in the past three years. The dataset is distributed with a license of GPL-2 and has a Kaggle usability score of 7.50. A unique point that makes this dataset stand out from other datasets available on the internet is the pictures taken are clear and the gestures in the pictures are standard.

Another dataset used to train the model is "ASL Alphabet" by Akash [7]. The dataset consists of 3001 RGB samples for each gesture. Even though it was updated 6 years ago, it could still be used now as the gestures for American Sign Language have not changed in the past six years. This model is also distributed with a license of GPL-2 with a Kaggle usability score of 8.75. A unique point that makes this dataset stand out from other datasets available is that the pictures are samples that were taken at diverse light conditions and complex backgrounds are included.

In this paper, the standard "small" would be defined as a dataset that consists of 1,000 samples or less per class. A "moderate" dataset is a dataset with less than 5,000 samples per class. A "large" dataset is defined as a dataset that contains more than 5,000 images per class.

Fig 2. Two samples from the final dataset of gestures representing "A" in ASL. The sample on the left is taken from Dataset A while the sample on the left is taken from Dataset B (Photo/Picture credit: Original).

Since both datasets are not considered small, only 150 samples in each gesture from each dataset are integrated into the combined dataset which will be used for training. The combined training set has 8842 samples in total, combined using the dataset above. An example of two samples from the final dataset representing "A" in American Sign Language is shown in Fig 2.

2.1.2 Data preprocessing

The images were loaded by using a target size of (224, 224, 3) (Using the form (x, y, z). The image loaded in is 244 pixels tall, 244 pixels wide, and has 3 dimensions.) and converted into array form using Keras. Next, the labels were encoded using the label encoder and were converted into numerical form to enable computational calculations. All input data was then preprocessed into the form MobileNet accepts.

2.2. Model Development

The model was transferred and fine-tuned to make it capable of detecting sign language alphabets by using Keras. Keras is a widely used high-level deep learning framework for artificial intelligence and machine learning. It acts as an interface that enables the opportunity for developers to create, train, and deploy neural networks easily and quickly. It also simplifies the process of designing and training neural networks by providing user-friendly API. Additionally, it is not only widely used in industry but also in research settings due to its adaptability and support for different neural network architectures. Most importantly, EfficientNet and MobileNet models are built-in models in Keras, allowing us to transfer the models with only one line of code.

2.2.1 Model selection

The EfficientNet model used was EfficientNetB0, with B0 representing the smallest variant of EfficientNet models. The reason for choosing this variant is that the training process would be more time-consuming if a larger model was used. A time-consuming training process contradicts one of the purposes of this investigation, which is testing the model's performance when there is limited time for development.

Two MobileNet models were tested, MobileNetV2 and MobileNetV3 Small. This is because both three models are lightweight, therefore the average training time of these models is acceptable and fulfils the purpose. However, MobileNetV3 Large is not selected to meet the purpose of this investigation, which is to simulate the environment where time constraints exist. Furthermore, MobileNetV1 is also not selected as MobileNetV1 is a relatively old technology that is not efficient enough to satisfy the requirement of this investigation.

2.2.2 Transfer learning and fine-tuning

Transfer learning is a machine learning technique in which the pre-trained models have been trained on a large dataset for general tasks are used and further tuned to allow the model to be able to complete a more specific job. Fine-tuning is adapting the network and allowing it to perform well in a specific situation.

Understanding the investigation's main aim is to test and compare the performance of two models, only the last output layer was removed and replaced with a Dense layer with SoftMax activation function.

2.2.3 Training environment

The computer configuration on which we trained the model is a MacBook Pro with Apple M1 chip, model number MacBookPro17, 1, and model identifier MYDA2CH/A. The computer is equipped with eight processor cores, including four performance cores and four efficiency cores, to ensure balanced performance under different workloads. The system memory is 8 GB, and the firmware version is 8419.60.4.

Its memory capacity of 8 GB may be somewhat limited for processing large-scale datasets and complex models. When working with large-scale data, a larger memory capacity usually improves training efficiency and processing speed.

In addition, while the Apple M1 chip has made significant advances in energy efficiency and performance, it may have some limitations compared to traditional high-performance computing platforms in certain areas of expertise and applications. Some specific areas of research and development may require more powerful computing resources, such as larger GPU memory or multi-GPU configurations, to meet demanding model training needs.

Overall, while the computer configurations on which we trained our models performed well in many application scenarios, more powerful hardware configurations may need to be considered for training needs when working with large-scale data and some domain-specific deep learning tasks.

2.3. Comparative Analysis

Two models will mainly be compared by using three criteria, training accuracy, testing accuracy, and training time. The testing-training percentage which represents the testing accuracy as a percentage of the training accuracy will also be calculated and compared as a final test for overfitting. If the percentage is over or equal to 100%, this shows that the model is not overfitting. Otherwise, the model is overfitting as the model performs less well on the testing dataset, which may be caused by learning unnecessary patterns.

2.3.1 Training accuracy

Training accuracy is a metric which measures the model's ability to fit the training data correctly. In simpler terms, it measures how well the machine learning performs on the training data. This is taken into consideration as it measures the accuracy of the model in the training process.

2.3.2 Testing accuracy

Testing accuracy is a metric which measures the model's ability to apply and recognise the samples in the test set. This is taken into consideration as it measures the accuracy of the model and provides an outline of how the model is likely to perform in the production environment.

If the testing accuracy of significantly higher than the training accuracy, the result that the model is overfitting can be deduced.

2.3.3 Training time

Training time is defined as the time required for the model to finish training. Understanding the purpose of this investigation is to test the model's viability and the ability to be fine-tuned to fit specific data, training time should also be taken into consideration. Subjecting to the scenario of the investigation, the training time is better if it is lower.

The training time of this investigation is the time used to train under the training environment mentioned in 2.2.3 Training environment.

3. Results

Table 1. presents the result of this investigation. In general, the results are not satisfactory in all three models. For MobileNetV3, the training accuracy is increasing as the epochs are increasing, while the training accuracy remains low consistently. This suggests overfitting. Test accuracy for MobileNetV2 is outstanding at first, but it later varies, raising questions regarding its stability and

generalisation. Like MobileNetV3, EfficientNetB0's test accuracy stays low, but its training accuracy increases across epochs. This also suggests an overfitting. All models show a notable training time, especially at higher epochs.

Table 1. Results of this investigation

Models	Training Accuracy of the Final Epoch (%)			Testing Accuracy (%)			Training Time (min)			Testing-Training Percentage (%)		
	5	10	20	5	10	20	5	10	20	5	10	20
MobileNetV2	38.29	52.83	41.27	61.08	4.61	39.23	59	64	164	159.51	8.73	95.06
MobileNetV3 (Small)	27.66	33.69	4.09	3.40	3.57	4.09	7.93	15	389	12.29	12.14	100
EfficientNetB0	54.69	56.67	66.53	5.22	3.57	3.57	54	108	1481	9.54	9.21	5.37

3.1. Training and Testing Accuracy of MobileNet Models

3.1.1 Training and testing accuracy of MobileNet V3

At epoch 5, MobileNet V3 has a training accuracy of 27.66% and a test accuracy of 3.40%.

As the epoch increases, the training accuracy improves (33.69% at epoch 10 and 34.92% at epoch 20), but the test accuracy remains low (3.57% at epoch 10 and 4.09% at epoch 20).

The relatively high training accuracy but low test accuracy may suggest overfitting. The model may have learned too many specific patterns from the training data and failed to generalise them to the unseen test data. However, the training time increases significantly, especially at epoch 20, where the training time reaches 389 minutes, and there may be excessive training time.

The weak improvement in test accuracy observed contradicts the hypothesis (Increasing epochs will substantially improve testing accuracy.). This discrepancy suggests a possible overfitting problem, thus calling into question the model's ability to generalise effectively.

3.1.2 Training and testing accuracy of MobileNet V2

At epoch 5, the training accuracy of MobileNet V2 is 38.29% while the testing accuracy is 61.08%.

As the epoch increases, the training accuracy is 52.83% at epoch 10 and 41.27% at epoch 20. The test accuracy is 4.61% at epoch 10 and 39.23% at epoch 20.

The gap between training and testing accuracy decreases, but the model still seems to perform poorly on the testing data. This may indicate that the model did not generalise sufficiently on the training data. The training time is relatively short, especially at epoch 5 and epoch 10. The training time for epoch 5 and epoch 10 are 59 and 64 minutes respectively. The training time they have even reached an astonishing 389 minutes at 20 epochs. However, it is still reasonable compared to other models.

While MobileNet V2's performance at epoch 5 was impressive, there was an unexpected drop in test accuracy at epoch 10, questioning the hypothesis that the model exhibits stable performance with a consistent increase in testing accuracy. This trend suggests the possibility of overfitting and emphasises the need for careful selection of epochs.

3.2. Training and Testing Accuracy of EfficientNetB0

At epoch 5, the training accuracy of EfficientNetB0 is 54.69% while the testing accuracy is 5.22%.

As the epoch increases, the training accuracy is 56.67% at epoch 10 and 66.53% at epoch 20. The test accuracy is 3.57% at epoch 10 and 3.57% at epoch 20.

The gap between training and testing accuracy is also large, and there may be some overfitting. The training time was relatively short, especially at epoch 5 and epoch 10, at 54 and 108 minutes, respectively. However, at epoch 20, the training time increased significantly to 1481 minutes, and there may have been excessive training time.

While training accuracy improved significantly over the epoch, test accuracy did not increase significantly, contradicting the hypothesis that EfficientNetB0 will demonstrate notable

improvements in both training and testing accuracy over epochs. This result highlights the importance of evaluating model generalisation and may require further research into architectural adjustments.

3.3. Summary

Taken together, the gap between the training and testing accuracies of the model can provide some clues about the model's performance and generalisation ability. Overfitting is a possible problem that can be mitigated by methods such as adding more data, using regularisation techniques, or adjusting the model structure. It is also important to keep an eye on whether the test accuracies stabilise after enough training epochs to better assess the model's performance. MobileNet V5 achieved a relatively high test accuracy in a relatively short training time, becoming the best group of this investigation. However, the final choice should also be weighed against specific tasks and resource constraints.

The results across architectures align with the general hypothesis (Small-scale datasets may lead to overfitting, impacting model performance in real-world scenarios), highlighting the challenges posed by small-scale datasets, particularly evident in MobileNet V3 and parts of MobileNet V2. Overfitting is observed, emphasising the need for caution in model development with limited data.

For data augmentation, the uniqueness of sign language gestures limits the application of standard data enhancement techniques, which is consistent with the hypothesis (Data augmentation might face challenges if there is an extreme lack of original data or when handling complex and sensitive datasets). This observation highlights the importance of tailored approaches in dealing with specialized datasets.

Hypothesis comparisons reveal subtle challenges associated with each architecture and dataset feature. Addressing these challenges requires improving training strategies, exploring more regularisation techniques, and considering the complexity of small datasets in sign language classification.

4. Discussion

4.1. Possible Factors

The observed performance results of the MobileNet V3, MobileNet V2, and EfficientNetB0 architectures on the Sign Language Alphabet dataset may be due to several factors. Some potential reasons for these results:

The small size of the dataset may not be sufficient to capture the variety of patterns and variations present in sign language gestures. Limited data may lead to overfitting, the model may memorize the training data instead of learning generalisable features.

The increased complexity of the MobileNet V3 and EfficientNetB0 architectures may lead to overfitting on smaller datasets. The increased capacity of these models may cause them to capture noise in the training data, thus reducing performance on unseen data.

The choice of the number of training Epoch is critical. The test accuracy of some models declines or plateaus at later epochs, which may indicate overfitting of the model. Choosing the optimal number of Epoch is critical to prevent the model from fitting too closely to the training data.

Suboptimal hyperparameter settings, such as learning rate, regularization strength or batch size, may affect model convergence and generalisation. Fine-tuning these hyperparameters may improve performance.

Sign language data itself may be sensitive to subtle changes, making the application of certain data enhancement techniques challenging. Careful consideration needs to be given when choosing an amplification method to avoid introducing noise.

4.2. Limitations of the Research

While this research provides a brief, detailed outlook for the difference in performance between MobileNet models and EfficientNetB0 while fitting small datasets, it is worth admitting that some limitations do exist in this research.

To begin with, the training platform used does not have sufficient computing power. Although it mocks the worst circumstance where a basic home-use laptop must be used, a lot of companies do have more powerful machines nowadays. Therefore, the experiment may be improved by using a more powerful machine.

Next, more models could be tested and compared. Due to computing power and time constraints, only a limited number of models were tested. This experiment could be improved by comparing more models such as EfficientNetB1 and MobileNetV1.

5. Conclusion

To conclude, most results are not too optimistic as the datasets are very small. It is not recommended to use the models tested on extremely limited datasets, especially for crucial jobs where accuracy is paramount.

However, MobileNetV2 with a training epoch of 5 performs unexpectedly well. It achieves 61.08% of testing accuracy and a testing-training percentage of 159.51%. MobileNetV2 may be considered when a small-sized dataset must be used.

To improve the performance of the model, future research could focus on acquiring more diverse and extensive datasets, experimenting with different data enhancement strategies, and fine-tuning hyperparameters. In addition, investigating alternative model architectures or combining migration learning with models pre-trained on larger datasets may provide insights into addressing the challenges posed by sign language alphabet classification.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning. PMLR, 2019: 6105-6114.

[2] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

[3] Jaryani F, Amiri M. A Pre-Trained Ensemble Model for Breast Cancer Grade Detection Based on Small Datasets. Iranian Journal of Health Sciences, 2023, 11(1): 47-58.

[4] Pramudhita D A, Azzahra F, Arfat I K, et al. Strawberry Plant Diseases Classification Using CNN Based on MobileNetV3-Large and EfficientNet-B0 Architecture. J. Ilm. Tek. Elektro Komput. dan Inform, 2023, 9(3): 522-534.

[5] Yang Y, Zhang L, Du M, et al. A comparative analysis of eleven neural network architectures for small datasets of lung images of COVID-19 patients toward improved clinical decisions. Computers in Biology and Medicine, 2021, 139: 104887.

[6] Kaggle. Kaggle: Your Home for Data Science Kaggle.com. 2022. 2024/1/10. https://www.kaggle.com/.

[7] Akash. ASL Alphabet//www.kaggle.com. 2024/1/10. https://www.kaggle.com/datasets/grassknoted/asl-alphabet.