American Sign Language Alphabet Recognition using Pretrained CNN Models: Comparison Between Fine-tuning and Feature Extraction

Yijing Chen
Department of Computer Science
University of Toronto
yijing.chen@mail.utoronto.ca

Ming Liu
Department of Computer Science
University of Toronto
gavin.liu@mail.utoronto.ca

Wenfei Wang
Department of Computer Science
University of Toronto
wenfei.wang@mail.utoronto.ca

Abstract

American Sign Language (ASL) has provided many deaf and speech-impaired individuals within Canada and US with much needed way to express their thoughts and feelings. However, for the people who have never learned ASL, it is difficult to understand without an interpreter. With the lack of a large, centralized dataset on ASL and the abundance of different pretrained Convolution Neural Network (CNN) models. One of the easiest implementation is to use transfer learning methods on the pretrained models to re-purpose their gained knowledge for this new classification problem. The goal of this work is to provide a comparison between the two of the most utilized transfer learning methods: fine-tuning and feature extraction with a variety of pretrained CNN models, to hopefully provide some guidelines on the advantages and disadvantages of these methods.

1 Introduction

For individuals who are deaf or have hard time hearing, sign language is often one of the best means for them communicate and express themselves. In the English-speaking region of Canada, the most widely used sign language is the American Sign Language (ASL) [15]. An obvious method to provide easy ASL interpretation is to utilize a deep learning solution on a large amount of ASL image data. However, unlike the more popular visual recognition classification problem, there are no large and complete ASL image dataset. Therefore, utilizing transfer learning with pretrained CNN models has been a popular approach[2, 14] in this area.

Some of the most widely used transfer learning methods includes fine-tuning and feature extraction, which is also used many times for ASL image recognition. On the other hand, few works have been published analyzing and comparing the advantages and disadvantages of the two methods. In this study, we use fine-tuning and feature extraction methods on 3 popular CNN models: InceptionV3, VGG16 and ResNet50, which are all pretrained on the ImageNet dataset, and compare their speed and accuracy on the ASL Alphabet dataset [1].

2 Related Works

Early sign language recognition are mainly based on HMMs. In 1995, Starner et al. [10] described a real-time HMM-based system for recognizing sentence level ASL. Recent years, the breakthrough of

Preprint. Under review.

deep learning provides a robust approach to generalize inter-subject variability and can consider the time-series dynamics behavior of human movements mainly with CNN.

In 2017, MD Rashedul Islam et al. [8] used deep Convolutional Neural Network (DCNN) for extracting efficient hand features to recognize the ASL using hand gestures and got presicion of 94.57%. And later in 2021, Barbhuiya et al. [2] applied CNN based feature extraction on modified VGG16 models for ASL classification and reached 99.82% accuracy.

There is no doubt that the accuracy of ASL recognition must be an important indicator to compare models, but at the same time, we also need to test on the speed of recognition. Most CNN models require a lot of memory and computation, the amount of calculation is an important concern of ASL recognition, especially in future development of real-time ASL recognition system. Therefore, we are testing and comparing the speed of the models as another criteria in this paper.

This paper applies deep learning based CNN for robust modeling of static signs in the context of ASL recognition, compares the accuracy and speed, and highlights the pros and cons of fine tuning and feature extraction. The evaluation shows that the models slightly breaks the limit of accuracy reported by current popular CNN models.

3 Method/Algorithm

Since the result from the 2 transfer learning methods may differ depending on the model used, we chose 3 of the most popular CNN widely used for a variety of image recognition tasks: Inception-V3, ResNet50 and VGG16.

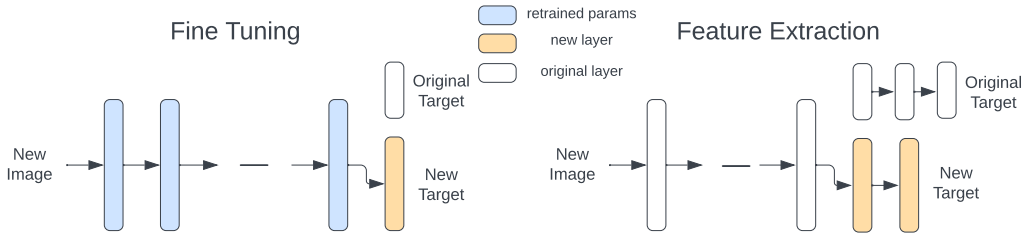

Figure 1: Comparison between Fine-tuning and Feature Extraction

3.1 Fine-tuning

Fine-tuning is one of the most popular method of transfer learning utilized by the computer vision community [5]. Often, a model pretrained on a very large dataset such as ImageNet is used and is fine-tuned using a smaller, more specific dataset to achieve a high-accuracy result [4]. The method involves with unfreezing some or all of the pretrained model, adding one or more fully connected layers to match the number of categories for classification, and training it with smaller number of data and a very small learning rate. Smaller learning rate is used since most of the pretrained weights do not need to have drastic changes for learning the new dataset [5].

3.2 Feature Extraction

As the name suggest, feature extraction utilizes the pretrained CNN model to extract features by taking the outputs from the activation of a hidden layer. A new model can than be trained by classifying these feature outputs [3]. Usually, the last hidden layer or the hidden layer close to the end are chosen for extract [7]. However, we think it is important for us to experiment with various layers. Therefore, for each of the 3 chosen CNN model, we chose 3 layers to extract feature from by examining the models' structure.

3.3 CNN

The original Inception network was introduced by Google in the ILSVRC 2014 Classification Challenge. It contains 22 layers in total with stacked inception modules, each containing varying

sizes of convolution filters ranging from 1x1 to 5x5 [11]. These filters are used to extract image features at varying levels. Inception-V3, introduced a year later, increased the number of layers to 42 and reduced the number of parameters, which decreased computation cost without decreasing result accuracy [11]. The inception-V3 model we used in the study is pretrained using the ImageNet dataset.

VGG was proposed by Karen Simonyan and Andrew Zisserman (2014) [9], which consisted of 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. A sequence of convolutional layers are followed by a max-pooling layer, and three fully connected layers are added at the end. The model we used in the study is also pretrained using the ImageNet dataset.

ResNet50 is a 50 layer version of the model ResNet, which is proposed by He et al. [6] in 2015. The model aims to solve the degradation problem brought by the increasing number of convolutional layers. The model introduces residual blocks, which includes skip connections to pass through information as an identity function. This approach has shown that it prevents the accuracy from degrading. Since ResNet learns residuals rather than actual image features, we think this models provides an interesting comparison to the other models. The ResNet50 model we used in the study is pretrained using the ImageNet dataset.

4 Experiments and Discussions

4.1 ASL Image Data Preparation and Augmentation

The dataset used is the ASL Alphabet dataset. It contains 87,000 coloured images from 29 classes [1]. The classes consists of 26 English alphabet letters as well as 3 special characters: SPACE, DELETE and NOTHING. Each image has a size of 200 x 200 pixels and depicts a static hand pose related to its class. Each class contains the hand gesture in various lighting condition and sizes, however the orientation of the hand stayed relatively similar.

Figure 2: Samples from ASL Alphabet dataset [Akash]

To introduce more variety, each image is randomly flipped and rotated in a range between -36 to 36 degrees. To reduce computation cost, we resized the image to 76 x 76 pixels before training the models.

4.2 Fine-tuning

All of the three pretrained models are modified in the same method for fine-tuning. We removed the last fully connected layer from the original model. Then added a global average pooling layer and a dense layer that maps to our classification dimension. For hyper parameters, we used a batch size of 32, cross entropy for loss function and Adam optimizer with learning rate 0.01. The models are first trained without updating the pretrained parameters with 30 epochs. Then, all parameters are trained with 10 epochs while changing the learning rate to 1e-5 for retaining more of the pretrained value.

4.3 Feature Extraction

Due to the consideration of fairness when comparing the two transfer learning method, the new model that is trained from the extracted feature has the same composition as the fine-tuning models, which

is a global average pooling layer and a dense layer that maps to our classification dimension. The hyper parameter used are also the same with training iteration of 30 epochs.

For the choices of the 3 feature extraction layers, we aim to spread out our choices across the entire model, as well as keeping in mind these models' unique architectures.

For inception-V3, we followed the models architecture and chose three hidden layers to extract output feature from. The first layer is the output from the third inception module (which is before the first reduction module). The second layer is the output from the forth inception module after the first reduction module (which is right before the second reduction module). The third layer is the second inception module after the second reduction module (before global average pooling) [11 12].

We extract the outputs from the 2nd, 3rd, and 4th max-pooling layer of VGG16. The max-pooling layers are choosen as they conclude the features extracted by previous convolutional layers.

For ResNet50, we chose the output of the third block of conv_2, the forth block of conv_3, and the third block of conv_5 [6]. These three layers represents the feature extraction result from the first convolutions, second convolutions and last convolutions.

4.4 Result and Discussion

First we compare the validation loss of each method:

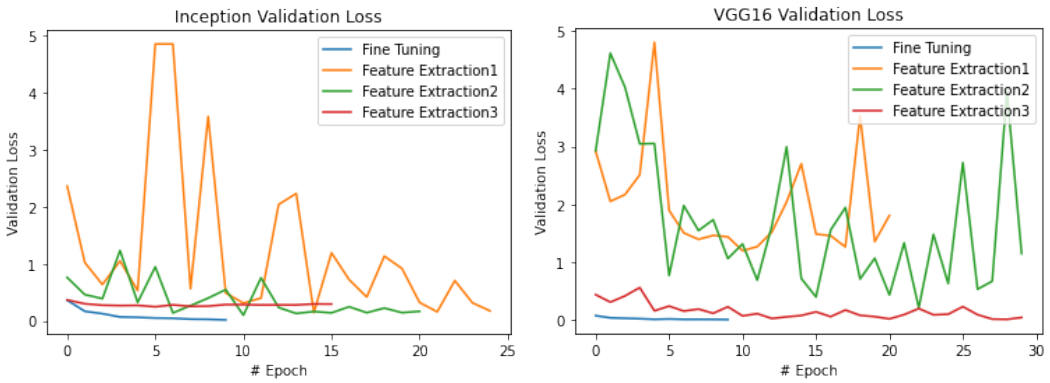

Figure 3: Validation Losses for Inception-V3 and VGG16 Model

For Figure 3 and 4, Feature Extraction 1, 2, and 3 are ordered from the shallowest hidden layers to the deepest hidden layers.

Observing Figure 3 and Figure 4, we can see that models that uses fine-tuning are always able to obtain the lowest validation loss. This result is expected, since for fine-tuning, the pretrained model is trained again in addition of the new layers/model added. However, the validation loss of the best feature extraction model is also not much higher than fine-tuning result.

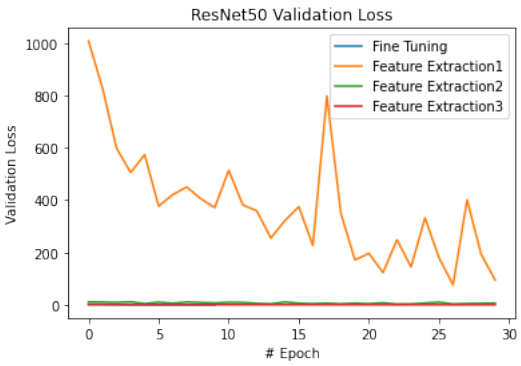

Figure 4: Validation Loss for ResNet50 Model

One important point to consider is that our dataset images and the categories are very different from the images these model are pretrained to. Thus, it may not always be the deepest layer that is able to best capture the features of ASL images. Looking at the validation loss graph of Inception-V3 in Figure 3, we can see that after 13 epochs, Feature Extraction 2 begins to perform better than Feature Extraction 3 and is not far from the fine-tuning result. Similarly, Feature Extraction 2 and 3 of ResNet50 performs identically to the fine-tuning result. For VGG16 the last hidden layer (Feature Extraction 3) has a performance very similar to fine-tuning after 10 epochs.

The reason why Inception-V3 and ResNet50 have better performance on shallower hidden layers compared to VGG16 may be because VGG16 contains less number of layers compared the other two models. This may suggest that VGG16's shallower layers failed to extract the most important features for identifying the hand gestures.

The test accuracy comparison of the 3 models are as follows:

	Fine Tuning	Feature Extraction 1	Feature Extraction 2	Feature Extraction 3
Inception v3	99.3	99.7	99.6	92.7
VGG16	99.9	71.6	94.0	99.6
ResNet50	94.0	57.5	66.5	80.7

Similar to the trend we see from the validation loss, for each model, the best performing feature extraction achieves very similar, if not better, accuracy compared to fine-tuning.

5 Conclusion

In this paper, we used ASL Alphabet dataset [1] to explore and compare two transfer learning methods, fine-tuning and feature extraction. The results from the experiment has shown that choosing a good hidden layer for feature extraction can achieve very similar (and sometimes even better) results to fine-tuning. Since feature extraction often requires less total computation cost since we don't need to retrain the pretrained method, in some situations it may be a better strategy to choose feature extraction on a good hidden layer than using fine-tuning.

For future extensions, it may be great to explore more varieties of pretrained model and attempt feature extraction on every possible layers to find a more precise best layer for feature extraction. Testing this with a variety of different new dataset may also be beneficial, since the difference between the pretrained and new dataset can possibly impact the choice of best feature extraction layers. In addition, there are many more transfer learning methods that we did not explore in this paper, such as joint training [13] and learning-without-forgetting [7]. Including more transfer learning method may help to discover even better choices for transfer learning.

References

[1] Akash. (2018). ASL Alphabet, Version 1. https://www.kaggle.com/datasets/grassknoted/asl-alphabet.

[2]Barbhuiya, A.A., Karsh, R.K. Jain, R. (2021). *CNN based feature extraction and classification for sign language*. Multimed Tools Appl 80, 3051–3069 . https://doi.org/10.1007/s11042-020-09829-y

[3] Donahue, J., Jia, Y. , Vinyals, O. , Hoffman, J. , Zhang, N. , Tzeng, E. and Darrell, T. (2014). *"Decaf: A deep convolutional activation feature for generic visual recognition,"* in International Conference in Machine Learning (ICML). https://arxiv.org/abs/1310.1531

[4] Girshick, R., Donahue, J., Darrell, T., amp; Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2014 IEEE Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2014.81

[5] Han, S., Pool, J., Tran, J., amp; Dally, W. J. (2015). *Learning both weights and connections for efficient neural networks*. https://arxiv.org/abs/1506.02626

[6] He, K., Zhang, X., Ren, S., amp; Sun, J. (2016). *Deep residual learning for image recognition*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90

[7] Li, Z., amp; Hoiem, D. (2017). *Learning without forgetting*. https://arxiv.org/abs/1606.09282

[8] M. R. Islam, U. K. Mitu, R. A. Bhuiyan and J. Shin. (2018). *"Hand Gesture Feature Extraction Using Deep Convolutional Neural Network for Recognizing American Sign Language,"* 2018 4th International Conference on Frontiers of Signal Processing (ICFSP). pp. 115-119, doi: 10.1109/ICFSP.2018.8552044.

[9] Simonyan, Karen, and Andrew Zisserman. *"Very deep convolutional networks for large-scale image recognition."* arXiv preprint arXiv:1409.1556 (2014).

[10] Starner, T. and Pentland, A. (1995). *Real-time American Sign Language recognition from video using hidden Markov models.* Proceedings of International Symposium on Computer Vision - ISCV, 1995, pp. 265-270, doi: 10.1109/ISCV.1995.477012.

[11] Szegedy et al. (2015) *Going Deeper with Convolutions.* CVPR 2015. IEEE Explore.

[12] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., amp; Wojna, Z. (2016). *Rethinking the inception architecture for computer vision*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.308

[13] Tompson, J. , Jain, A. , LeCun, Y. , & Bregler, C. (2014, September 17). *Joint training of a convolutional network and a graphical model for human pose estimation*. arXiv.org. Retrieved April 22, 2022, from https://arxiv.org/abs/1406.2984

[14] Töngi, R. (2021, February 25). Application of transfer learning to sign language recognition using an inflated 3D deep convolutional neural network. arXiv.org. Retrieved April 22, 2022, from https://arxiv.org/abs/2103.05111

[15] U.S. Department of Health and Human Services. (2019) *American Sign Language*. National Institute of Deafness and Other Communication Disorders. https://www.nidcd.nih.gov/health/american-sign-language

Contribution

5.1 Report

Yijing Chen, Ming Liu, Wenfei Wang,

5.2 Data preparation and Augmentation

Ming Liu

5.3 Building Experiment Model

Ming Liu, Wenfei Wang, (Yijing Chen tried the proposal CNN with LSTM but didn't get any result that meets the expectations)

5.4 Experiment Testing

Ming Liu, Wenfei Wang