

# The Impact of Combining Two Filters on Image Denoising and How to Analyze the Results

1<sup>st</sup> Wenhao You

*Department of Computing Science*  
*University of Alberta*  
Edmonton, Canada  
wyoul@ualberta.ca

2<sup>nd</sup> Leo Chang

*Department of Computing Science*  
*University of Alberta*  
 Edmonton, Canada  
 basu@ualberta.ca

**Abstract**—XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
XX  
XX  
XX  
XX  
XX

**Index Terms**—convolutional neural networks, frequency regularization.

## I. INTRODUCTION

Currently, people can not live without mobile devices. These compact yet powerful gadgets have become indispensable tools for communication, entertainment, and information. Their portability and versatility make them a constant companion. Meanwhile, Convolutional neural networks play an important role in computer vision applications. However, these neural networks are usually implemented on high-specification hardware. There are several advantages of running convolutional neural networks on mobile devices: privacy, internet, and runtime. For enhancing privacy, personal information does not need to be uploaded or transmitted to the cloud servers anymore. For reducing the dependence on the internet connection, the functionality on local devices can replace some internet services. For the runtime, especially some applications that need real-time feedback, without connecting to the cloud server can shorten the processing time. In all, convolutional neural networks can totally replace the usage of many applications on mobile devices, ensuring personal data security.

According to the popularity of mobile devices and the benefits of convolutional neural networks, we want to find a way to deploy some large and complex convolutional neural networks on mobile devices, leading to the question: “How can we deploy large convolutional neural networks on mobile devices?”

We found five methods to achieve our goal: upgrade the hardware of mobile devices; use Extreme Learning Machine (ELM) [1] to allocate the weight of the hidden layers randomly in order to train large models on mobile devices faster; use NestDNN [2] dynamically adjust the size and computational complexity of the network based on available resources on mobile devices; use “One-shot Whole Network Compression” [3] to prune, quantize, and compress the neural networks; use Frequency Regularization (FR) [4] to reduce parameters by removing high-frequency component.

After conducting a thorough literature review, considering all the limitations, accuracy, complexity, and future potential, we choose Frequency Regularization (FR) as our target algorithm, deploying it on one of the most popular mobile devices - Android mobile. Our main idea uses FR to compress a convolutional neural network U-Net and then decompresses the uploaded compressed model on an Android mobile device in a short time. After that, use the decompressed model to do image segmentation for the Carvana Image Masking Challenge Dataset [5].

The proposed main idea can be divided into three directions to achieve the final goal respectively:

- Direct deployment and tuning of FR code [6] on Android: Instead of introducing an additional operating system layer, this method focuses on deploying the Frequency Regularization (FR) algorithm within the Android ecosystem directly. The core idea of this direction is to optimize and adjust the FR parameters specifically for the Android hardware and software architecture.
- Deployment of Linux environment on Android system: It aims to add another layer to the Android system, providing a more controlled and flexible development environment for deploying and testing deep learning models. We choose Termux [7] which is an Android terminal application and Linux environment to deploy.
- Deployment of FR code on Android Studio: This method aims to optimize the FR algorithm for Android's native architecture, ensuring seamless integration and operation within Android. The main idea of this method is similar to the first direction above.

## II. RELATED WORK

XXX

### III. METHOD

XXXX

## IV. RELATED WORK

XXX

## REFERENCES

- [1] A. Akusok, L. Leal, K.-M. Björk, and A. Lendasse, “High-performance elm for memory constrained edge computing devices with metal performance shaders,” in *Proceedings of ELM2019*, ser. Proceedings in Adaptation, Learning and Optimization, J. Cao, C. Vong, Y. Miche, and A. Lendasse, Eds. International: Springer, 2021, pp. 79–88.
- [2] B. Fang, X. Zeng, and M. Zhang, “Nestdnn: Resource-aware multi-tenant on-device deep learning for continuous mobile vision,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom ’18. ACM, Oct. 2018. [Online]. Available: <http://dx.doi.org/10.1145/3241539.3241559>
- [3] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, “Compression of deep convolutional neural networks for fast and low power mobile applications,” 2016.
- [4] C. Zhao, G. Dong, S. Zhang, Z. Tan, and A. Basu, “Frequency regularization: Reducing information redundancy in convolutional neural networks,” *IEEE Access*, vol. 11, pp. 106 793–106 802, 2023.
- [5] M. M. M. P. W. C. Brian Shaler, DanGill, “Carvana image masking challenge,” 2017. [Online]. Available: <https://kaggle.com/competitions/carvana-image-masking-challenge>
- [6] G. Dong, “Frequency regularization,” <https://github.com/guanfangdong/pytorch-frequency-regularization>, 2023.
- [7] termux, “Termux application,” <https://github.com/termux/termux-packages>, 2023.