A Novel lightweight Convolutional Neural Network – ExquisiteNetV2

JOU, Shyh-Yaw, SU, Chung-Yen

Abstract

In the paper of ExquisiteNetV1, the ability of classification of ExquisiteNetV1 is worse than DenseNet.

In this article, we propose a faster and better model ExquisiteNetV2. We conduct many experiments to evaluate its performance. We test ExquisiteNetV2, ExquisiteNetV1 and other 9 well-known models on 16 credible datasets including Cifar-10 under the same condition. According to the experimental results, ExquisiteNetV2 gets the highest classification accuracy over half of the datasets. Important of all, ExquisiteNetV2 has fewest amounts of parameters. Besides, in most instances, ExquisiteNetV2 has fastest computing speed.

Model Architecture

ExquisiteNetV2 mainly consists of 4 types of blocks, namely DFSEB V2 block (DFSEB1 version 2), ME block[1], FCT block and EVE block. Besides, we design a new kind of SE block[2], that is, SE-LN block.

A. EVE Block

Complete name is extreme-value-expansion block.

In the research of ExquisiteNetV1[1], we found using max pooling will get better accuracy than using depthwise convolution with its stride set to 2. We are very curious about why max pooling also work well despite it discard the 75% features when downsampling.

According to Fig 1, after applying max/min pooling to a cute Corgi image, we can find that the result image of max pooling look brighter and the one of min pooling look darker. It is reasonable because max/min pooling will only retain big/small pixel values. Important of all, we can find that despite max/min pooling will discard 75% feature, the both result images still look much like original image, in other words, the feature retained by max/min pooling might still be able to represent the original feature.

To make ME block of ExquisiteNetV1 be able to retain more feature when downsampling, we combine a min pooling layer into ME block. So the amount of discard feature down to 50%. Because revised ME block retains the extreme feature, we rename it as EVE block. The architecture of EVE block is shown in Fig 2.

We also make a try to combine average pooling layer into EVE block to retain more feature. However, after adding the average pooling layer into EVE block, the accuracy become worse.

B. FCT Block

Complete name is Feature-Concentrator block.

The feature in the raw image is the most important so we should try to retain feature of raw image as many as possible.

In order to retain all feature of raw image, we combine the EVE block with a 4x4 depthwise conv and use FCT as input block of ExquisiteNetV2. The architecture of FCT block is shown in Fig 3.

C. SE-LN Block

Complete name is SE-Layer-Normalization block.

SE block include 2 fully-connected layer, we use layer normalization to replace these 2 fully-connected layer for decrease amount of parameters. The amounts of parameters of layer normalization is much fewer than ones of 2 fully-connected layer[3]. The architecture of SE-LN block is shown in Fig 4.
D. DFSEBV2 Block

We delete one SE block and replace the other SE block into SE-LN block so the amounts of parameters is fewer than DFSEBV1. Besides, we change some activation. The architecture of DFSEBV2 block is shown in Fig 5.

Fig. 5 DFSEBV2 block

E. ExquisiteNetV2

We show the architecture in Table. 1.

We only replace the first and second ME block in ExquisiteNetV1 with FCT and EVE block. One reason is that the amounts of parameters in ME block is fewer than FCT or EVE block, the other reason is that we think the raw feature is the most important so we only use FCT or EVE block in the front part of model. Besides, we change the pointwise conv in ExquisiteNetV1 into depthwise conv. The amounts of parameters of ExquisiteNetV2 is fewer than ExquisiteNetV1. The parameters of each model is shown on Table. 2.

Table. 2 the information of DTD

|                  | Original | Use in experiment |
|------------------|----------|-------------------|
| Training set     | 1880     |                   |
| Test set         | 1880     |                   |
| Classes          | 47       |                   |

2. Chest X-Ray

This dataset is used to recognize whether the children have the lung disease.

Table. 3 the information of Chest X-Ray

|                  | Original | Use in experiment |
|------------------|----------|-------------------|
| Training set     | 5232     |                   |
| Test set         | 624      |                   |
| Classes          | 2        |                   |

3. OCT

OCT is Optical Coherence Tomograph. The classes are diabetic macular edema, choroidal neovascularization, drusen and normal. The size of training set is so big that we randomly sample 250 images from every class.

Table. 4 the information of OCT

|                  | Original | Use in experiment |
|------------------|----------|-------------------|
| Training set     | 207130   | 1000              |
| Test set         | 1000     |                   |
| Classes          | 4        |                   |

4. Leukemia

This dataset is used to recognize whether the children have Leukemia.

Table. 5 the information of Leukemia

|                  | Original | Use in experiment |
|------------------|----------|-------------------|
| Training set     | 10661    |                   |
| Test set         | 1867     |                   |
| Classes          | 2        |                   |

5. RAF-DB

Every image is human’s facial expression. There are 2 sets in the original dataset, one is basic, the other is compound. We train and test on basic.

Table. 6 the information of RAF-DB

|                  | Original | Use in experiment |
|------------------|----------|-------------------|
| Training set     | 12271    |                   |
| Test set         | 3068     |                   |
| Classes          | 7        |                   |

6. SUN397

The dataset contains various scene images.

Table. 7 the information of SUN397

|                  | Original | Use in experiment |
|------------------|----------|-------------------|
| Training set     | 19850    |                   |
| Test set         | 19850    |                   |
| Classes          | 397      |                   |

7. Fruits-360

The size of training set is so big that we randomly sample 30 images from every class.

Table. 1 Architecture of ExquisiteNetV2

|                  | Operator | Out Channels |
|------------------|----------|--------------|
| Input            | 224×3    | FCT 12       |
|                  | 112×12   | DFSEBV2 12   |
|                  | 112×12   | EVE 48       |
|                  | 56×48    | DFSEBV2 48   |
|                  | 56×48    | ME 96        |
|                  | 28×96    | DFSEBV2 96   |
|                  | 28×96    | ME 192       |
|                  | 14×192   | DFSEBV2 192  |
|                  | 14×192   | ME 384       |
|                  | 7×384    | DFSEBV2 384  |
|                  | 7×384    | Depthwise Conv 384 |
|                  | 7×384    | Hard Swish 384 |
|                  | 7×384    | Average pooling 384 |
|                  | 1×384    | Dropout 384  |
|                  | 1×384    | FC -         |

Dataset

We choose 16 credible datasets in the experiment, namely DTD, Chest X-Ray, OCT, Leukemia, RAF-DB, SUN397, Fruits-360, Food-101, Stanford Online Products, Caltech256, STL-10, Birds-2011, VGG Flowers, Plant SeedlingsV2, Synthetic Digits and Cifar-10 [4]-[18].

1. DTD

It is a texture dataset. The resolution of every image range from 300×300~640×640.
8. Food-101
The size of training set is so big that we randomly sample 50 images from every class.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 22688 | |
| Training set | 67692 | 3930 |

Table. 8 the information of Fruits-360

9. Stanford Online Products
The size of training set is so big that we randomly sample 300 images from every class.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 25250 | |
| Training set | 75750 | 5050 |

Table. 9 the information of Food-101

10. Caltech256
The name is “256”, but there is actually 257 classes in the dataset. The 257th class is “clutter”. We think the 257th class is so unclear that we delete this class. The original dataset is not divide into training set and test set, so we divide each class into 10 equal parts. One of the parts is used as training set, the rest are test set.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 257 | 26899 |
| Training set | 30607 | 2881 |

Table. 10 the information of Stanford Online Products

11. STL-10
The dataset is inspired from CIFAR-10[19]. Because the resolution of the image in CIFAR-10 is too low(32x32) [19], the images in STL-10 are all 96x96.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 8000 | |
| Training set | 5000 | 2881 |

Table. 11 the information of Caltech256

12. Birds-2011
The dataset includes various bird species.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 5994 | |
| Training set | 5794 | 200 |

Table. 13 the information of Birds-2011

13. VGG Flowers
The dataset includes various flower species.

14. Plant SeedlingsV2
The dataset contains 12 kinds of crop seed. Because the original dataset is not divide into training set and test set, so we divide each class into 2 equal parts. One is used as training set, the other is test set.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 22688 | 2773 |
| Training set | 5539 | 2766 |

Table. 15 the information of Plant SeedlingsV2

15. Synthetic Digits
Every image in this dataset is RGB and their background is complicated.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 2000 | |
| Training set | 10000 | 10 |

Table. 16 the information of Synthetic Digits

16. Cifar-10
Every image is 32x32. Many paper use this dataset as benchmark. We randomly split 10000 image from training set as validation set.

| Classes | Original | Use in experiment |
|---------|----------|-------------------|
| Test set | 10000 | |
| Training set | 50000 | 10 |

Table. 17 the information of Synthetic Digits

**Experiment and Analysis**

A. *Settings*
To make the experimental result being objective, we test each model under the same conditions. The conditions are shown below:

1. Program is run on Ubuntu 16.04
2. Accelerator is GeForce RTX 2080 Ti
3. Every image is converted to RGB
4. Every image is resized to 224x224
5. Every pixel value in image is scaled into [0,1]
6. Each dataset is not augmented (only cifar-10 augment)
7. Every model is trained without pretrained weight, each weight in each layer is initialized by Pytorch default method.
8. Every model is test with 2 different optimizer respectively. One is RangerVA[21], the other is SGD[22].
9. We have set random seed so the images are read in the same order by each model
10. Batch size is 50 (cifar-10 is 64)
11. Initial learning rate is 0.05 and the factor is 0.1. If the training loss is not improved for consecutive 5 epochs, the
learning rate will be multiplied by factor. Training loss is the average loss of each image in the training set.
12. To avoid the overfitting phenomenon, we add the Dropout layer before the last fully-connected layer into each model.
13. End condition is training loss < 0.01 or total epochs reach 150. The reason why we design this way is that it is difficult to make training loss lower than 0.01 for some models.
14. We have test whether the SE-LN block is able to replace the SE block. We set the decrease ratio of each SE block to 3

B. Classification Results
We test many models, namely SE-ResNet18[2], MobileNetV3-Large, EfficientNet-b0, DenseNet121, ResNet18, ResNet50, ShuffleNetV2, GhostNet[23][28]. “LN-” means the all SE blocks in the original model are replaced by SE-LN blocks.

Table. 18 parameters of each model (when the last FC layer contains 1000 output channels)

| Model                  | Parameters (M) |
|------------------------|-----------------|
| DenseNet121            | 7.9788          |
| EfficientNet-b0        | 5.3             |
| MobileNetV3-L          | 5.483           |
| ShuffleNetV2           | 7.3939          |
| ResNet50               | 25.557          |
| ResNet18               | 11.6985         |
| SE-Resnet18            | 12.1515         |
| LN-Resnet18            | 11.6933         |
| GhostNet               | 5.1825          |
| SqueezeNet             | 1.2354          |
| SE-ExquisiteNetV1      | 1.5618          |
| LN-ExquisiteNetV1      | 1.3329          |
| SE-ExquisiteNetV2      | 1.0285          |
| LN-ExquisiteNetV2      | 0.8993          |

Table. 20 Test Acc RangerVA(II)

| Optimizer: RangerVA | Acc (%) |
|---------------------|---------|
| RAF                 | 70.57   |
| SUN397              | 36.89   |
| Fruit               | 98.26   |
| Food                | 19.14   |
| DenseNet121         | 70.57   |
| EfficientNet-b0     | 69.98   |
| MobileNetV3-L       | 68.25   |
| ShuffleNetV2        | 66.17   |
| ResNet50            | 65.68   |
| ResNet18            | 68.35   |
| SE-Resnet18         | 68.74   |
| LN-Resnet18         | 67.21   |
| GhostNet            | 67.41   |
| SE-ExquisiteNetV1   | 67.99   |
| LN-ExquisiteNetV1   | 65.87   |
| SE-ExquisiteNetV2   | 69.49   |
| LN-ExquisiteNetV2   | 70.08   |

Table. 21 Test Acc RangerVA(III)

| Optimizer: RangerVA | Acc (%) |
|---------------------|---------|
| Online              | 38.25   |
| Caltech256          | 20.09   |
| STL10               | 72.89   |
| Birds               | 38.66   |
| DenseNet121         | 38.25   |
| EfficientNet-b0     | 33.62   |
| MobileNetV3-L       | 36.12   |
| ShuffleNetV2        | 29.87   |
| ResNet50            | 30.45   |
| ResNet18            | 33.65   |
| SE-Resnet18         | 34.57   |
| LN-Resnet18         | 35.00   |
| GhostNet            | 30.10   |
| SE-ExquisiteNetV1   | 37.58   |
| LN-ExquisiteNetV1   | 38.93   |
| SE-ExquisiteNetV2   | 41.58   |
| LN-ExquisiteNetV2   | 43.13   |

Table. 22 Test Acc RangerVA(IV)

| Optimizer: RangerVA | Acc (%) |
|---------------------|---------|
| Flower              | 40.23   |
| Seed                | 94.95   |
| Synthetic Digits    | 98.30   |
| EfficientNet-b0     | 29.74   |
| MobileNetV3-L       | 26.26   |
| ShuffleNetV2        | 19.48   |
| ResNet50            | 28.70   |

Fig. 6 classification speed of models with different batch size
ExquisiteNetV2 has strong ability of classification. ExquisiteNetV2 is very near 1, it can be seen that LN
suitable to ExquisiteNetV2 model to classify the test set of RAF.

| Model       | Test Acc (%) |
|-------------|--------------|
| DenseNet121 | 8.79         |
| EfficientNet-b0 | 8.45      |
| MobileNetV3-L | 8.61       |
| ShuffleNetV2  | 8.54        |
| ResNet50     | 7.96         |
| ResNet18     | 7.89         |
| SE-ResNet18  | 7.88         |
| LN-ResNet18  | 7.87         |
| GhostNet     | 7.86         |
| SE-ExquisiteNetV1 | 8.75    |
| LN-ExquisiteNetV2 | 8.61     |

Note: LN-ExquisiteNetV2 is the model with the best performance in this comparison.

C. Analysis

From Table 23, we can learn the parameters of LN-ExquisiteNetV2 is fewest. From Fig., we can learn that as long as the batch size is bigger than 10, the LN-ExquisiteNetV2 is the fastest. The speed is calculated from how long for every model to classify the test set of RAF-DB.

From Table 24 to Table 25, we can learn that SGD is more suitable for ExquisiteNetV2. LN-ExquisiteNetV2 gets first place on 8 datasets when the optimizer is RangerVA and gets first place on 10 datasets when the optimizer is SGD. Besides, LN-ExquisiteNetV2 is much better than state-of-the-art models on some datasets. From Table 26, the average ranking of LN-ExquisiteNetV2 is very near 1, it can be seen that LN-ExquisiteNetV2 has strong ability of classification.

However, SE-LN block seems to be only suitable for ExquisiteNet especially ExquisiteNetV2.

D. Result on Cifar-10

| Model       | Val Acc (%) | Test Acc (%) |
|-------------|-------------|--------------|
| LN-ExquisiteNetV2 | 93.29       | 92.52        |

Table 28 Cifar-10 result of ExquisiteNetV2 (Train without pretrained weight)
**E. Object Detection**

YoloV5 is one of the best object detection models [29]. YoloV5 contains 4 different size, namely YoloV5s, YoloV5m, YoloV5l, YoloV5x. We combine ExquisiteNetV2 with YoloV5 and test it on Face Mask dataset [30]. Face Mask dataset is originally not divide into training set and test set so we choose image No. 0-49 as test set, the rest is training set.

From Table. , we can learn ExquisiteNetV2-YoloV5 has the fastest computing speed and fewest parameters. The ability of detecting objects of ExquisiteNetV2-YoloV5 is better than YoloV5 but the parameters of ExquisiteNetV2-YoloV5 is 20 M less than YoloV5. The FPS is test on nvidia Geforce 3060, the size of image is 320x320.

|                | mAP 0.5   | Parameter (M) | FPS  |
|----------------|-----------|---------------|------|
| YoloV5s        | 0.9308    | 7.07          | 172  |
| YoloV5m        | 0.9474    | 21.06         |      |
| YoloV5l        | 0.9739    | 46.64         |      |
| YoloV5x        | 0.9691    | 87.26         |      |
| ExquisiteNetV2-YoloV5 | 0.9434 | **0.95** | **208** |

Fig. 7 result on the test set of Face Mask

**Conclusion**

ExquisiteNetV2 has outstanding ability of analyzing data, fast computing speed, few parameters compared with state-of-the-art models. However, ExquisiteNetV2 is not good at analyzing the very low resolution images. Important of all, according to the experimental results, the performance of ExquisiteNetV2 completely beat ExquisiteNetV1.

**References**

[1] S. Y. Zhou and C. Y. Su, "Efficient Convolutional Neural Network for Pest Recognition-ExquisiteNet," 2020 IEEE Eurasia Network Conference on IOT, Communication and Engineering (ECICE), pp. 216-219, 2020.

[2] J. Hu, L. Shen, S. Albanie, G. Sun and E. Wu, "Squeeze-and-Excitation Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 8, pp. 2111-2025, 1 Aug. 2020, doi: 10.1109/TPAMI.2019.29193327.

[3] J. L. Ba, J. R. Kiros, and G. E. Hinton. “Layer normalization,” arXiv preprint, arXiv:1607.06450, 2016.

[4] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed and A. Vedaldi. "Describing Textures in the Wild," Proceedings of the IEEE Conf. on CVPR, 2014.

[5] K. Daniel, Z. Kang, G. Michael (2018), “Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images,” Mendeley Data, V3, doi: 10.17632/rscbjbr9sij.3.

[6] A. Gupta, & R. Gupta, (2019). ALL Challenge dataset of ISBI 2019. The Cancer Imaging Archive. https://doi.org/10.7937/tcia.2019.dc6446r

[7] Li, Shan, Deng, WeiHong and Du, JunPing, “Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild," IEEE CVPR, pp. 2584-2593, 2017.

[8] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. “SUN Database: Large-scale Scene Recognition from Abbey to Zoo,” IEEE CVPR.

[9] H. Muresan and M. Oltean, “Fruit recognition from images using deep learning,” Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.

[10] B. Lukas, G. Matthieu and V. G. Luc, “Food-101 -- Mining Discriminative Components with Random Forests,” ECCV, 2014.

[11] H. O. Song, Y. Xiang, S. Jegelka and S. Savarese. “Deep Metric Learning via Lifted Structured Feature Embedding.” In IEEE CVPR, 2016.

[12] G. Griffin, AD. Holub, P. Perona, “The Caltech 256,” Caltech Technical Report.

[13] A. Coates, H. Lee, A. Y. Ng. “An Analysis of Single Layer Networks in Unsupervised Feature Learning.” AISTATS, 2011.

[14] Alex Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” “2009.

[15] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, “The Caltech-UCSD Birds-200-2011 Dataset,” Computation & Neural Systems Technical Report, CNS-TR-2011-001.

[16] M-E. Nilisback, and A. Zisserman, “Automated flower classification over a large number of classes,” Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, 2008.

[17] T. M. Giselson, R. N. Jørgensen, P. K. Jensen, M. Dyrmann, and H. S. Midiby, “A public image database for benchmark of plant seedling classification algorithms,” arXiv preprint arXiv:1711.05458, 2017.

[18] R. Prasun, G. Subhankar, B. Saumik and P. Umapada, “Effects of Degradations on Deep Neural Network Architectures,” arXiv preprint arXiv:1807.10108, 2018.

[19] Alex Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” “2009.

[20] Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, 2009.

[21] H. Yong, J. Huang, X. Hua, and L. Zhang. “Gradient centralization: A new optimization technique for deep neural networks,” 2020, arXiv preprint arXiv:2004.01461.

[22] H. Robbins, S. Monro “A Stochastic Approximation Method,” The Annals of Mathematical Statistics, Vol. 22, No. 3., pp. 400-407, 1951.

[23] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, “Searching for mobilenetv3,” CoRR, abs/1905.02244, 2019.

[24] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," arXiv preprint arXiv:1905.11946, 2019.

[25] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” arXiv, abs/1608.06993, 2016.

[26] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE CVPR, 2016, pp. 770-777, doi: 10.1109/CVPR.2016.90.

[27] N. Ma, X. Zhang, H. Zheng, and J. Sun, “Shufflenet V2: practical guidelines for efficient CNN architecture design,” CoRR, vol. abs/1807.11164, 2018.

[28] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, “GhostNet: More Features from Cheap Operations,” arXiv:1911.11907 [cs], Nov. 2019.

[29] Ultralytics, “YoloV5,” doi: 10.5281/zenodo.4679653.

[30] Misc. “Mask Dataset,” Make ML.

Kaggle: https://www.kaggle.com/andrewmvd/face-mask-detection