IOP Conf. Series: Journal of Physics: Conf. Series **1487** (2020) 012027 doi:10.1088/1742-6596/1487/1/012027

Generalization and Visual Comprehension of CNN Models on Chromosome Images

Chengyu Wang[1], Daiyun Huang[1], Jionglong Su[2], Limin Yu[3] and Fei Ma[2,*]

[1] Department of Electrical Engineering and Electronics, University of Liverpool based in Xi'an Jiaotong-Liverpool University, Suzhou, China
[2] Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou, China
[3] Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China

[*]Email: fei.ma@xjtlu.edu.cn

Abstract. Computer-aided image classification has achieved start-of-the-art performance since Convolutional Neural Network structures were employed. Classical neural networks such as AlexNet and VGG-Net inspired several rules of designing network models. Besides benchmark datasets such as MNIST, CIFAR and ImageNet, classification performance of medical images such as chromosome karyotyping images also improved via Convolutional Neural Network. However, there are few studies on generalization among different datasets. In this paper, we designed a neural network with nine layers, and achieved classification accuracy of 0.984, 0.816 and 0.921 on the dataset of MNIST, CIFAR and chromosome karyotype images. We also visualized the output of several layers of the model and explained that smooth output between neural network layers may induce lower accuracy on classification.

1. Introduction

Classification with Convolutional Neural Network (CNN) architecture has achieved states-of-the-art performance [1] [2][3] [4] [5] in several benchmark datasets [6] [7]. With regards to medical images such as chromosome karyotyping images, there are also several researches based on CNN [8] [9] [10]. Chromosome karyotyping is used to diagnose several diseases such as [11] [12] [13]. LeCun et al. studied a Neural network (LeNet) with five layers [1] based on gradient descent and achieved the accuracy beyond 0.98 on MNIST dataset; Alex et al. proposed an eight layers Neural Network (AlexNet) [2] won the champion of ILSVRC 2012, which employed ideas of Rectified Linear Unit (ReLU), Local Response Normalization (LRN), dropout and max pooling; VGG-Net from Oxford Visual Geometry Group [4] proposed in 2014 and achieved the state-of-the-art performance, use smaller kernel with size 3 x 3 instead of 11 x 11 or 5 x 5 which used in AlexNet. VGG-Net also verified a deeper layer (16) Neural Network may generalizing better among different datasets. Wen et al. studied a CNN structure on chromosome dataset and achieved accuracy of 0.937 [14]; Qin et al. proposed two-stage strategy and finally achieved accuracy of highest accuracy per patient case of 0.992 [15];

In this paper, we revised a Neural Network named ANet-c6, which expended the depth of convolutional layers of AlexNet. After uniformed data of three datasets: MNIST, CIFAR and Chromosome karyotype images, ANet-c6 get the better results on both three datasets comparing to

AlexNet and VGG-Net. We also visualized and analysed outputs of several layers of neural network models.

2. Methodology

We trained three CNN networks on three datasets, by visualizing the specific output of layers, we show the potential reason for their performance. One of three models is named ANet-c6 borrowed from AlexNet, which consistent of six convolutional layers, five pooling layers and three fully connected layers, shown as following **Figure 1**.

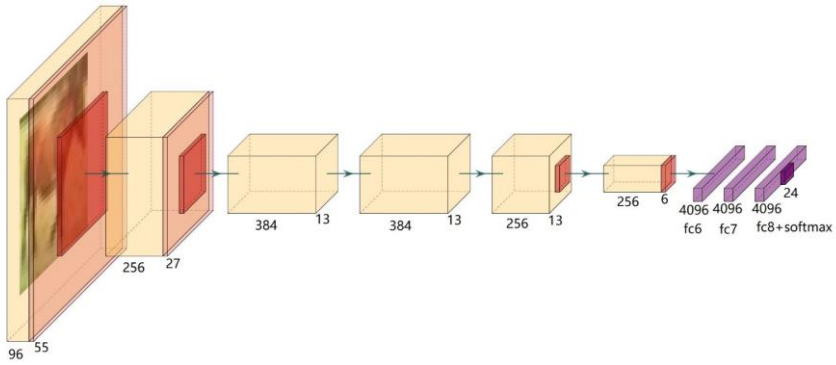


Figure 1. Structure of ANet-c6 model. There is a LRN (Local Response Normalization) layer after convolutional layer one and two. After pooling layer five, there is a sixth convolutional layer and pooling layer succeeded. For Softmax layer, output are 10 labels for MNIST and CIFAR, while 24 labels for Karyotype.

Other two models are AlexNet and VGG-Net, which proposed in 2012 and 2014 espectively.

2.1 Dataset preparation

For the sake of comparison, we handled the datasets by following steps:

2.1.1 Extract jpg images from original dataset. For MNIST dataset, we use ‘mnist’ module lib from TensorFlow to extract image data and label from four compressed files, after walk through four files, all 60000 images were saved as ‘jpg’ files, with the file name denoting the corresponding label.

For CIFAR dataset, we first use ‘pickle’ module from Python lib to load all six pickle files, which consists of five training data and one testing data. Secondly we extracted each pickle file with four elements dictionary information consists of batch number, label, image data and filename. Thirdly we walk through all image data, swap the dimension from [0,1,2] to [1,2,0]. This can convert image data from ‘png’ file format (which CIFAR data compressed with) to ‘jpg’ file format (which will be consistent with MNIST and Karyotype dataset).

For single chromosome karyotype images in ‘tif’ format, we walk through all images and convert them to ‘jpg’ files.

2.1.2 Wrap images for training, validating and testing. We set parameters of ratio for validating and testing as 5% and 5% respectively for all three datasets.

Then we wrap all ‘jpg’ images into ‘tfrecords’ files, which is suggested by official documents. The information of wrapped ‘tfrecords’ file including: label, image data, original image size (height, width and channel) and filename.

Note there are three ‘tfrecords’ files for each data set: train, valid and test.

2.2 Training, validating and testing

2.2.1 Early break. Besides relaying on the architecture of the models to reduce overfitting, we also

employ the early-break strategy. To be precisely, we checking the accuracy on validating dataset after each epoch, if the accuracy beyond validating expectation, and the training accuracy also beyond the training expectation for five consecutive batch, the training procedure will break and finish.

2.2.2 Check and adjust. A common used strategy of setting learning rate is scale down the value after a certain number of iteration. We proposed 'check and adjust' algorithm to decrease the learning rate, as shown in following Algorithm 1.

Algorithm 1.

Input: *acc1*: the accuracy of this mini-batch; *acc_best*: the best accuracy from all mini-batch
Output: *lr*: learning rate for training, which change according to *gate* value;
need_save: indicate if need save training parameters; *acc_best*: best accuracy currently.
Parameters: *level 1/2/3/4*: the accuracy level which used to set different *gate* and *lr* value; *gate 1/2/3/4*: the number which used to decide if need to save current training parameters by checking the difference value of *acc1* and *acc_best*.

1) if *acc_best* > level 1/2/3/4:
2) set *gate* 1/2/3/4 and *lr*
3) if (*acc1* - *acc_best*) > *gate*
4) set *need_save* and *acc_best* to *True* and *acc1*
5) return

2.3 Layer output visualization

For the sake of understanding the differences between final performance of models employed in this paper, we extracted the output of several layers via 'tensorboard', and saved them as images. By comparing the differences between different layers and different models, we found some possible explanation.

3. Experiments

The hardware of server using in this paper is a Sugon workstation, equipped with four NVIDIA GPUs of 2080ti, two Intel CPUs of Xeon Silver 4110, and 64GB memory.

The operating system of the server is Ubuntu 18.04.03 LTS, TensorFlow 1.14.0 and python 3.6.8 are employed as software develop.

3.1 Dataset

MNIST and CIFAR datasets are public datasets can be accessed from internet [16] [17], while the chromosome karyotyping dataset is offered by a company in Suzhou city, China. Single chromosome images were extracted from pre-processed file, which each file consists 46 single ones. We will NOT explain more details on how to extracted and pre-processed procedure in this paper as they are not relevant much to the topic. Three datasets are employed as following **Table 1**:

Table 1. Sample number and size of each dataset in the experiments of this paper. Note that the size of images in Karyotype dataset are not equal among all samples, most of them are between 45 x 57 and 96 x 198

	Total samples	Training samples	Validating samples	Testing samples	Sample size
MNIST	70000	62584	3550	3596	28 x 28
CIFAR	60000	54022	3025	2953	32 x 32
Karyotype	89980	80944	4415	4621	45 x 57 to 96 x 198

Examples from each dataset is as following **Figure 2**:

(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 2. examples from dataset: (a) and (b) from MNIST; (c) and (d) from CIFAR; (e) and (f) from Karyotype.

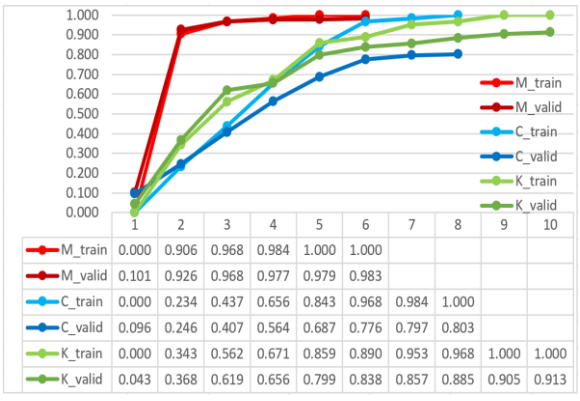
In view of AlexNet performance better with input image size of 227 x 227 than 224 x 224, we resize feed data to AlexNet and ANet-c6 as 227 x 227 as well, and VGG-Net fed with 224 x 224.

All the samples fed to neural networks are wrapped into 'tfrecords' file, which shuffle the samples and output mini-batch images.

3.2 Training, validating and testing

In consideration of the size of datasets and the depth of CNN network architecture in this paper are not so consuming, we run three models on three different GPUs separately, each model using a single GPU.

In training procedure, we set the mini-batch size to 64, epoch number to 200. One step stands for once mini-batch training, and accuracy evaluation are implemented every 200 steps via **Algorithm 1.** Note the action of saving training parameter as 'ckpt' file only occur (if needed) after this evaluating operation, or every 10 epochs regardless of training accuracy evaluation. Since we employed 'early break' strategy, all the training procedure are break before run over all epochs which supposed to be 200. 'early break' is reasonable for reduce overfitting, as shown in following **Figure 3**, from which we can find the dark green line (validating) growing smoothly below the light green line (training), denoting gets the highest accuracy.



	1	2	3	4	5	6	7	8	9	10
M_train	0.000	0.906	0.968	0.984	1.000	1.000				
M_valid	0.101	0.926	0.968	0.977	0.979	0.983				
C_train	0.000	0.234	0.437	0.656	0.843	0.968	0.984	1.000		
C_valid	0.096	0.246	0.407	0.564	0.687	0.776	0.797	0.803		
K_train	0.000	0.343	0.562	0.671	0.859	0.890	0.953	0.968	1.000	1.000
K_valid	0.043	0.368	0.619	0.656	0.799	0.838	0.857	0.885	0.905	0.913

Figure 3. Accuracy of training and validating on three datasets with ANet-c6 model. As the model's check point depends on every 200 steps, training accuracy on MNIST dataset grow faster than other two, hence only one 'ckpt' files saved before training accuracy beyond 0.9, while on CIFAR and Karyotype dataset, model saved 5 and 6 'ckpt' files respectively. Note the first file is saved after the model just run just one mini-batch and we simply set the accuracy to '0' if it is smaller than 0.2.

In validating procedure, we loaded all the saved training parameters (ckpt files), run them on the validating dataset, and picked one with highest validating accuracy. The validating result on three datasets with 'ANet-c6' models are shown as **Figure 3** above.

In Testing procedure, we fed test dataset to models with best validating parameters (from 'ckpt' file)

and get the testing accuracy.

Overall accuracy of three models on three datasets are shown in following **Table 2**, from which we find ANet-c6 performance better than other two models on all three datasets.

Table 2. Accuracy of three models on three datasets. 'M', 'C', and 'K' are short for datasets of 'MNIST', 'CIFAR' and 'Karyotype'.

	Train on M	Validate on M	Test on M	Train on C	Validate on C	Test on C	Train on K	Validate on K	Test on K
AlexNet	1.0	0.982	0.982	0.984	0.804	0.804	1.0	0.911	0.910
ANet-c6	1.0	0.983	**0.984**	1.0	0.803	**0.816**	1.0	0.913	**0.921**
VGG-Net	0.984	0.983	0.980	1.0	0.646	0.670	1.0	0.879	0.884

3.3 Output visualization

From testing results, we found testing accuracy on CIFAR with VGG-Net model was rather lower (0.67) than other two models on all three datasets. We extracted the output of two samples from three datasets by ANet-c6 model and VGG-Net model respectively, shown as following **Figure 4**. We found that comparing to ANet-c6, VGG-Net extracted more smooth output after convolutional layer one (**conv1**), pooling layer one (**pool1**) and pooling layer two (**pool2**). As shown in (a-12, b-12, c-12) vs (a-22, b-22, c-22). However, we noticed the output after final pooling layer (**pool_**) of VGG-NET was too distinguishable as they are "too smooth" (d-22), which means might lost too much details. We will do further research on modifying the last pooling layer of VGG-Net to improve its performance.

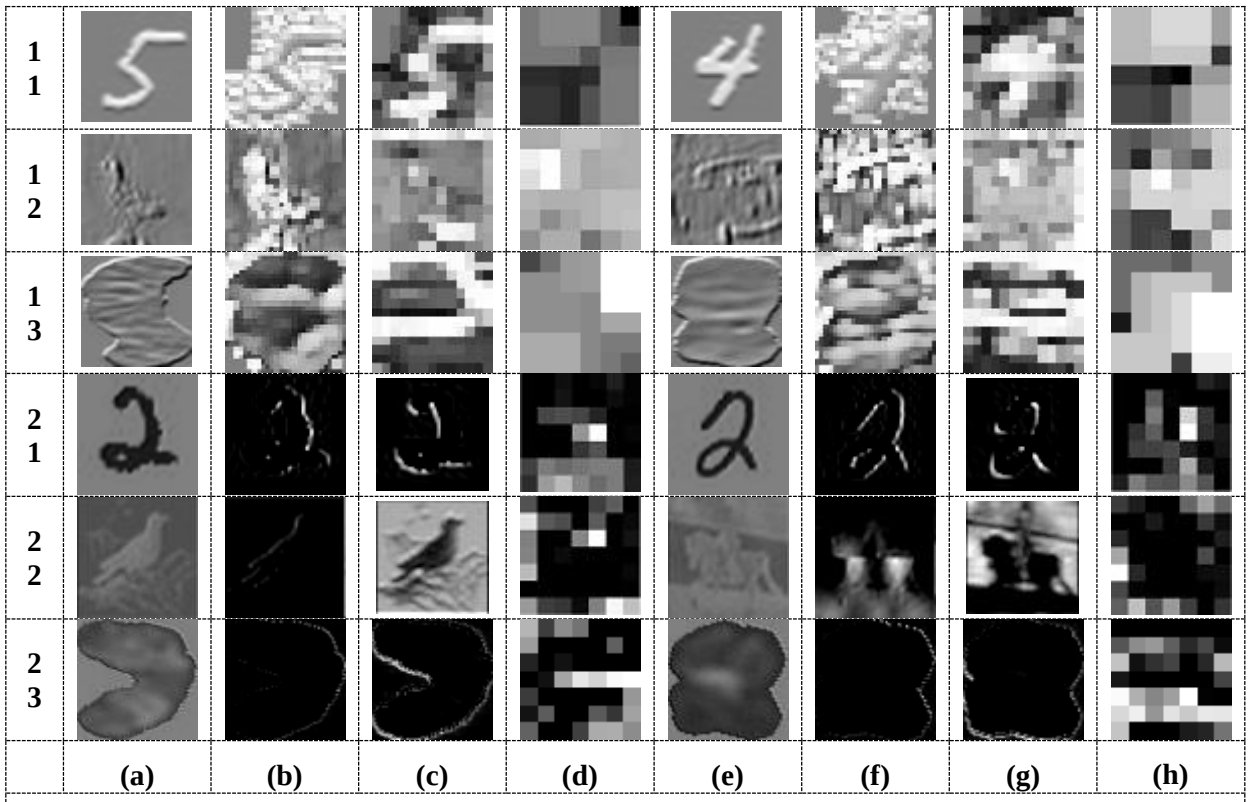


Figure 4. layer output visualization. Each row contains two samples from a dataset. Row name x1, x2, x3 indicates sample from dataset **MNIST (x1)**, **CIFAR (x2)** and **Karyotype chromosome (x3)**, where x can be **1 (ANet-c6)** or **2 (VGG-NET)**. For example, row 11 stands for results from ANet-c6 on MNIST, and row 22 means results from VGG-Net on CIFAR. Each row contains two samples, each sample corresponding four columns. Columns (a) to (d) are for first sample, they are outputs from

conv1, pool1, pool2 and pool_; columns (e) to (h) are for second sample, they are outputs from same layers as first sample. For instance, (a)-12 is conv1 output of ANet-c6 on first sample of CIFAR, and (a)-22 is conv1 output of VGG-Net on first sample of CIFAR. Note that samples from same dataset are not identical for different models.

4. Conclusion

We proposed a neural network named ANet-c6 which consists of six convolutional layers and three fully connected layers. We also uniformed three datasets, first two datasets are benchmark datasets, and the third one is a dataset of chromosome karyotype images. ANet-c6 achieved better generalization accuracy on all three datasets than AlexNet and VGG-NET.

By comparing the testing results and outputs of several layers of ANet-c6 and VGG-Net, we identified the possible reasons which influence the final testing results. We assume that VGG-NET did not work as well as expected due to last pooling layer or not fine-tuned hyper-parameters, which can be improved. With regards to the visualization, a special designed structure [3] may also be used to reveal what a layer does in generalizing procedure. This is also our research direction in the future.

Acknowledgement

This study is supported by the funds of Xi'an Jiaotong-Liverpool University:

1) Research Development Fund: RDF-17-02-51;
2) Key Programme Special Fund (KSF-P-02);
3) Construction of a Bioinformatics Platform for Precision Medicine: RDS10120180041;

Reference

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 1, Lake Tahoe, NV, USA, Dec 2012, pp. 1097–1105.

[3] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in Computer Vision – ECCV 2014, Zurich, Switzerland, 2014, pp. 818–833.

[4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in 3rd International Conference on Learning Representations, San Diego, CA, USA, May 2015.

[5] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, Feb 2017, pp. 4278–4284.

[6] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141–142, Nov 2012.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, and Ieee, "Imagenet: A large-scale hierarchical image database," in IEEE-Computer-Society Conference on Computer Vision and Pattern Recognition Workshops, ser. IEEE Conference on Computer Vision and Pattern Recognition, 2009, Conference Proceedings, pp. 248–255.

[8] X. Hu, W. Yi, L. Jiang, S. Wu, Y. Zhang, J. Du, T. Ma, T. Wang, and X. Wu, "Classification of metaphase chromosomes using deep convolutional neural network," Journal of Computational Biology, vol. 26, no. 5, pp. 473–484, MAY 1 2019.

[9] Swati, G. Gupta, M. Yadav, M. Sharma, and L. Vig, "Siamese networks for chromosome classification," in 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, Oct 2017, pp. 72–81.

[10] T. Arora and R. Dhir, "Correlation-based feature selection and classification via regression of segmented chromosomes using geometric features," Medical & Biological Engineering & Computing, vol. 55, no. 5, pp. 733–745, MAY 2017.

[11] X. Wang, Development of a computer-aided chromosome analysis system to assist cancer diagnosis, 2008.
[12] D. A. Griffiths, "Shifting syndromes: Sex chromosome variations and intersex classifications," Social Studies of Science (Sage Publications, Ltd.), vol. 48, no. 1, pp. 125–148, 2018.
[13] B. B. Ganguly, D. Banerjee, and M. B. Agarwal, "Impact of chromosome alterations, genetic mutations and clonal hematopoiesis of indeterminate potential (chip) on the classification and risk stratification of mds," Blood Cells, Molecules and Diseases, vol. 69, pp. 90–100, MAR 2018.
[14] W. Zhang, S. Song, T. Bai, Y. Zhao, F. Ma, J. Su, and L. Yu, "Chromosome classification with convolutional neural network based deep learning," in 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, Oct 2018, pp. 1–5.
[15] Y. Qin, J. Wen, H. Zheng, X. Huang, J. Yang, L. Wu, N. Song, Y. Zhu, and G. Yang, "Varifocal-net: A chromosome classification approach using deep convolutional networks," IEEE Transactions on Medical Imaging, pp. 1–1, 2019.
[16] Y. LeCun, "The mnist database," Website, accessed 30 Nov. 2019. http://yann.lecun.com/exdb/mnist/
[17] A. Krizhevsky, "The cifar-10 dataset," Website, accessed 30 Nov. 2019. http://www.cs.toronto.edu/ kriz/cifar.html?usg=ALkJrhjqbhW2lLxLo8EmqNS-tbK0aT96JQ