A Pilot Study of Novel Multi-Filter CNN Layer

Author: Mohamed Aboukhair
maak8991@gmail.com

Abstract

Convolutional Neural Networks (CNNs) reached their peak of complex structures, but until now few researchers have addressed the problem of relying on one filter size. Mainly a 3×3 filter is the most common one which is being used in any structure. Only at the first layers of the CNN model, filters bigger than 3x3 could be partially used. Most of the researchers work with filters (size, values, etc) as a blackbox. To the best of our knowledge, no prior work has opened this box. Our research is the first pilot study which proposes a new multi-filter layer in which different filters with variant sizes are used. Our proposed multi-filter layer aims to create a strong learning model while avoiding the risk of both the exponential high training time and the overfitting problem.

Keywords: CNN , CNN structures , Classification

1 Introduction

1.1 what is the problem?,what is significant ?

A convolutional neural network (CNN, or ConvNet) is a type of artificial neural networks (ANNs), most commonly used to analyze or classify visual imagery.[1] A convolutional layer is the backbone of building CNN which extracts features on the basic and complex level of processing images inspired from human brain [2; 3]. The convolutional layer is consisted of several filters , those filters can detect various features and it is improving in the training stage. The filters can be consider as the most important part of CNN. Although the filter of CNN is important , we heavily depend on one size of filter as main size of all filters. It is not standard to use 3×3 filters, as researchers try to explore the effects of different filter sizes.[4; 5; 6; 7] It is only recommended to use a 3x3 filter size due to three reasons: First, it has lower parameters to adjust in the training stage which speeds up the training process.[4; 6] Second, it does not support over-fitting because of its size it lowers the chance of memorizing the data.[7] Third, the need for a higher depth CNN model makes it harder to use bigger filters.[4; 6] Bigger filters are used in many standard models partially to enhance the performance like the 7×7 filter at the first layers of the CNN model.

1.2 are there any solutions?

Researchers have experimentally proved that as the filters increase in size, the learning process will get slower, the overfitting will highly occur and the complexity on the model will increase to find better weights. This lead the researchers to lower their focus on filter size effects due to proven facts and stop exploring variations of filter size experiments.

1.3 what is limitation ?

To the best of our knowledge, No research has been found about using a single layer with a multi-filter sizes. All other researchers discuss using multiple filters separately with multiple layers or they discuss the effects of different sizes of filters on CNN learning phase. this shows the lack of exploring the filter size experiments. Hardware limitation is also reason for not exploring bigger filters due to the need of higher computational power to overcome the time complexity[8].

1.4 what we want to achieve ?

We aim to open the black-box of the CNN layer by analyzing the effects of different filter sizes and also by studying the usage of multiple filter sizes in the same layer. We also apply a different percentage of each filter size in the same layer to avoid heavily using bigger filter sizes while keeping part of the strong learner. These new structures use bigger filters to create more valuable features which are higher in terms of quality for the CNN to enhance its performance. We avoid using too much bigger filters to avoid the need of higher computational power and the need of more parameters to adjust with acceptable time to skip the exponential increase in time complexity.[8]

The contributions of this paper are

- A novel Multi-Filter CNN layer.
- A novel CNN structures based on Multi-Filter layer.
- An exploration analysis of Multi-Filter CNN layer advantages and disadvantages .

2 Previous Literature

2.1 intro

As far as we know , the concept of using multiple filters in same layer does not exist in previous work of researchers yet the analysis of filter-size variants exists.We are going to discuss the analysis as it has been the main inspiration to the Multi-Filter CNN layer.

2.2 mush up or state each research work

Y. Camg ozl u and Y. Kutlu as well as O. Khanday, S. Dadvandipour, and M. A. Lone [4; 6] showed analysis of different filter sizes effects which we noticed 3x3 filter based models has far the best results yet a combination of different filter size could get better results than 5x5 and 9x9 filter based models. Also they shows the impact of filter size on computational power and time complexity which also w. Ahmed and A. Karim [5] support as well as showing both the effects of filter sizes on different image sizes and the exponential increased of time complexity in models that use bigger filter sizes.they also show that bigger filter sizes can get similar results as 3x3 filter based models in some cases. those cases could be explained by Ozturk, U. Ozkaya, B. Akdemir, and L. Seyfi [7] whom stated the convolution filter 3x3 and 7x7 both seem to be more successful than 5x5 and 9x9 convolution filter which memorize the data due to having the ability to perform the learning process strongly. these findings suggest that Large scale filters have encountered the problem of overfitting with the problem of the exponentially increase in time complexity.

2.3 conclusion of review

This could suggest using bigger filter can have benefits yet it is unusable if benefits is smaller than the disadvantages. Firstly, the exponential increased of time complexity problem has been introduce then the strong learner problem which could lead to overfitting. Those problems are main wall that prevent researchers from opening the black-box of filter-size and trying new experiments on bigger filter sizes or trying to solve these problems to acquire the potential advantages of strong learner.

3 Data and Methods

3.1 Dataset

Datasets has been chosen carefully to exploit the new layer advantages and discover possible disadvantages. we design couple of rules needed to find certain datasets.

Flower Classification with TPUs

this is kaggle competition which has goal of classifying 104 types of flowers based on their images drawn from five different public datasets. this competition is now mainly used in getting starting with TPU because of its big size.[9]

Reason

- big dataset to show the effects of the proposed layer
- variety of input image size to test effect of our proposed layer on three image sizes from same dataset.
- 100 classes to show the effect of the proposed layer on complex globalization problem

ISIC 2018 HAM10000

The HAM10000 data-set, a large collection of multi-source dermatoscopic images of common pigmented skin lesions, Training of neural networks for automated diagnosis of pigmented skin lesions is hampered by the small size and lack of diversity of available dataset of dermatoscopic images. They tackle this problem by releasing the HAM10000 ("Human against Machine with 10000 training images") dataset [8].

Classes \ Compare by	Num.	percent.%
MEL	1113	11.1
NV	6705	66.9
BCC	514	5.1
AKIEC	327	3.3
BKL	1099	11.0
DF	115	1.1
VASC	142	1.4
Sum	10015	100

Table 1: HAM10000 Data-set Table

Reason

- medium dataset to see effects of poropsed layer on common problems
- huge imbalance to decide whatever the proposed layer is affected by such problem
- 7 classes with too many Venn pattern to show the effect of complex features that are repeated in different classes

SIIM-ISIC Melanoma Classification

this is competition which melanoma in images of skin lesions will be identified. In particular, the problem is to classify whatever patient has melanoma or not. this competition is continuation of ISIC 2018 HAM10000.[10] Reason

- binary dataset to see if our proposed layer is affected by binary problems
- huge imbalance to decide whatever the proposed layer is affected by such problem within binary problem

3.2 Multi-Filter CNN layer

Multi filter layer: Multi-filter layer is a combination of a certain number of layers with different filter sizes each which output the same dimension as normal CNN layer. The proposed layer has two problems: the first problem is the output dimension that cannot be equal with different filter sizes, the second problem is computational power needed to run bigger filters and the third problem is overfitting due to using stronger learners. The main idea is to solve those problems within the Multi-filter layer to make it more usable in real time problems. Firstly, we solve the output dimension by using a padding function to fill the empty spaces in the desired output dimension by zero. Secondly, we solve the time complexity problem by defining a lower percentage factor. The bigger filter-size gets the lower percentage factor of its existence in the Multi-filter layer. Finally, we solve the stronger learner problem which may lead to overfitting by increasing 3x3 filter size percentage above 50

3.3 Multi-Filter CNN Structures

The Multi-filter layer is just a layer which can be used in any CNN structure to enhance the model. There are many variants for the usage of Multi-filter layers which can be only limited by machine learning engineering but we are going to show two simple structures due to the unlimited variants of the Multi-filter layer structures.

Fixed Structure

Fixed structure is based on replacing each layer that is only based on 3x3 filter size by Multi-filter layer with fixed percentage for each filter size that exists within the Multi-filter layer. There are certain recommendations for choosing the each filter size percentage of existence within The Multi-filter layer.

Recommendations (maybe in the discussion as reveled)

- Filter size 3x3 must exist with the chosen filter sizes (time complexity limitation)
- Filter size 3x3 must have the dominating percentage (time complexity limitation)
- The bigger filters size gets the smaller percentage assigned to it(time complexity limitation + overfitting)
- Max filter sizes that can be used to replace layers of filter size 3x3 in famous structures like ResNet or DenseNet are 7x7 filter sizes.

Decreasing Structure (DS)

Decreasing structure Same concept as fixed structure yet instead of using fixed percentage of filter size, we use decreasing equation which decreases percentage of bigger filters based on position of layer in the structure. On the other side the 3x3 filter size percentage increases by the value that has been taken from the bigger filter sizes.

Possible advantages: Lowering time complexity by lowering bigger filter sizes percentage which lead to lowering the parameters (weights) that need to be adjusted. Lowering overfitting by lowering bigger filter sizes percentage in the lower the existence of stronger learners in the upper layers of the structure.

4 Results

5 Discussion

6 Conclusion

7 Acknowledgments

References

[1] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, and N. Chervyakov, “Application of the residue number system to reduce hardware costs of the convolutional neural network implementation,” *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378475420301580

[2] K. Fukushima, S. Miyake, and T. Ito, “Neocognitron: A neural network model for a mechanism of visual pattern recognition,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 5, pp. 826–834, 1983.

[3] H. Aghdam and E. Heravi, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*, 01 2017.

[4] Y. Camgözlü and Y. Kutlu, “Analysis of filter size effect in deep learning,” *CoRR*, vol. abs/2101.01115, 2021. [Online]. Available: https://arxiv.org/abs/2101.01115

[5] W. Ahmed and A. Karim, “The impact of filter size and number of filters on classification accuracy in cnn,” pp. 88–93, 04 2020.

[6] O. Khanday, S. Dadvandipour, and M. A. Lone, “Effect of filter sizes on image classification in cnn: a case study on cfir10 and fashion-mnist datasets,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, p. 872, 12 2021.

[7] S. Ozturk, U. Ozkaya, B. Akdemir, and L. Seyfi, “Convolution kernel size effect on convolutional neural network in histopathological image processing applications,” in *2018 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*. IEEE, Nov. 2018. [Online]. Available: https://doi.org/10.1109/isfee.2018.8742484

[8] L. Alzubaidi, J. Zhang, A. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, 03 2021.

[9] M. McDonald, M. Görner, M. Görner, P. Bailey, P. Bailey, and Y. G. Phil Culliton, “Flower classification with tpus,” 2020. [Online]. Available: https://kaggle.com/competitions/flower-classification-with-tpus

[10] A. Zawacki, B. Helba, G. Shih, J. Weber, J. Elliott, M. Combalia, N. Kurtansky, NoelCodella, P. Culliton,

Models \ Approaches	normal	75-20-5	88-10-2	85-15	95-5	DS	Model Mean
Resnet50	0.702692	0.729641	0.70263	0.713319	0.692082	0.723663	0.7106712
Resnet101	0.671153	0.684769	0.692851	0.682038	0.680302	0.686797	0.682985
Resnet152	0.642984	0.671854	0.650927	0.660868	0.665423	0.659332	0.6585647
Densnet201	0.733961	0.764336	0.753442	0.763077	0.746641	0.774995	**0.7560754**
Densnet169	0.721506	0.745419	0.736944	0.753119	0.730462	0.753319	0.7401282
Densnet121	0.680174	0.730526	0.690994	0.692649	0.678247	0.716134	0.6981207
apporach Mean	0.6920784	**0.72109**	0.7046314	0.710845	0.698859	0.71904	0.7077576

Table 2: 512x512 Flower Classification with TPUs

Models \ Approaches	normal	75-20-5	88-10-2	85-15	95-5	DS	Model Mean
Resnet50	0.637421	0.687875	0.677682	0.6708	0.663736	0.656819	0.6657222
Resnet101	0.63336	0.636421	0.627173	0.629113	0.642139	0.647832	0.6360064
Resnet152	0.603214	0.616144	0.617201	0.605956	0.621056	0.590569	0.6090234
Densnet201	0.753493	0.759805	0.755706	0.757587	0.752007	0.762643	**0.7568735**
Densnet169	0.747237	0.756493	0.741352	0.747237	0.746772	0.76157	0.7501102
Densnet121	0.71484	0.732462	0.720554	0.714883	0.700778	0.740248	0.7206275
apporach Mean	0.6815942	**0.6982**	0.6899447	0.687596	0.687748	0.6932802	0.6897272

Table 3: 331x331 Flower Classification with TPUs

Models \ Approaches	normal	75-20-5	88-10-2	85-15	95-5	DS	Model Mean
Resnet50	0.59899	0.57727	0.599228	0.587141	0.597094	0.594936	0.5924432
Resnet101	0.541079	0.535817	0.555112	0.560715	0.548555	0.523029	0.5440512
Resnet152	0.524456	0.51835	0.48366	0.502581	0.531513	0.509719	0.5117132
Densnet201	0.724654	0.73695	0.72609	0.746029	0.725778	0.73636	**0.7326435**
Densnet169	0.728132	0.722181	0.721289	0.732845	0.72488	0.733254	0.7270969
Densnet121	0.707247	0.727366	0.708872	0.709329	0.713754	0.73147	0.7163397
apporach Mean	0.6374264	0.63632	0.6323752	0.639773	**0.64026**	0.638128	0.6373813

Table 4: 224x224 Flower Classification with TPUs

Models \ Approaches	normal	75-20-5	88-10-2	85-15	95-5	DS	Model Mean
Resnet50	0.77297	0.786285	0.768975	0.789614	0.774967	0.758323	0.775189
Resnet101	0.774967	0.790946	0.776299	0.769641	0.773636	0.773636	0.776521
Resnet152	0.770307	0.759654	0.77763	0.770307	0.768975	0.762317	0.768198
Densnet201	0.762317	0.788949	0.769641	0.747004	0.772304	0.787617	0.771305
Densnet169	0.762983	0.781625	0.779628	0.771638	0.774301	0.766978	0.772859
Densnet121	0.782957	0.793609	0.776965	0.786285	0.768975	0.79028	0.783179
apporach Mean	0.771083	0.783511	0.774856	0.772415	0.772193	0.773192	0.774542

Table 5: 224x224 ISIC 2018 Task 1

and V. Rotemberg, "Siim-isic melanoma classification," 2020. [Online]. Available: https://kaggle.com/competitions/siim-isic-melanoma-classification

Models \ Approaches	75-20-5	88-10-2	85-15	95-5	DS
Resnet50	2.695	-0.0063	1.0628	-1.061	2.0972
Resnet101	1.3616	2.1698	1.0885	0.9149	1.5644
Resnet152	2.887	0.7944	1.7884	2.2439	1.6348
Densnet201	3.0375	1.9481	2.9116	1.268	4.1034
Densnet169	2.3913	1.5439	3.1613	0.8956	3.1813
Densnet121	5.0352	1.0821	1.2475	-0.1928	3.596
mean	2.90127	1.25534	1.87669	0.6781	2.69619

Table 6: 512x512 Flower Classification percentage difference

Models \ Approaches	75-20-5	88-10-2	85-15	95-5	DS
Resnet50	5.0454	4.0261	3.3379	2.6315	1.9398
Resnet101	0.3061	-0.6187	-0.4247	0.878	1.4472
Resnet152	1.2931	1.3987	0.2743	1.7843	-1.2645
Densnet201	0.6312	0.2214	0.4095	-0.1486	0.915
Densnet169	0.9257	-0.5885	0	-0.0465	1.4333
Densnet121	1.7622	0.5715	0.0044	-1.4062	2.5409
mean	1.66062	0.83509	0.60024	0.61542	1.16862

Table 7: 331x331 Flower Classification percentage difference

Models \ Approaches	75-20-5	88-10-2	85-15	95-5	DS
Resnet50	-2.172	0.0239	-1.1849	-0.1896	-0.4054
Resnet101	-0.5262	1.4034	1.9636	0.7477	-1.805
Resnet152	-0.6107	-4.0796	-2.1876	0.7057	-1.4737
Densnet201	1.2296	0.1436	2.1376	0.1125	1.1706
Densnet169	-0.5951	-0.6843	0.4714	-0.3252	0.5123
Densnet121	2.012	0.1625	0.2082	0.6508	2.4224
mean	-0.1104	-0.50509	0.23472	0.28365	0.0702

Table 8: 224x224 Flower Classification percentage difference

Models \ Approaches	75-20-5	88-10-2	85-15	95-5	DS
Resnet50	1.3316	-0.3995	1.6645	0.1998	-1.4648
Resnet101	1.5979	0.1332	-0.5327	-0.1332	-0.1332
Resnet152	-1.0653	0.7324	0	-0.1332	-0.799
Densnet201	2.6632	0.7324	-1.5313	0.9987	2.53
Densnet169	1.8642	1.6645	0.8656	1.1319	0.3995
Densnet121	1.0653	-0.5992	0.3329	-1.3982	0.7324
mean	1.24282	0.3773	0.13317	0.11097	0.21082

Table 9: 224x224 ISIC 2018 Task 1 percentage difference