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 that 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 of the model will increase to find better weights. This leads 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 the limitation ?

To the best of our knowledge, No research has been found about using a single layer with 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 the CNN learning phase. this shows the lack of exploring the filter size experiments. Hardware limitation is also the reason for not exploring bigger filters due to the need for 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 CNN to enhance its performance. We avoid using too much bigger filters to avoid the need for higher computational power and the need for 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.
- 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

To the best of our knowledge, the utilization of multiple filters within a single layer has not been explored in prior research. However, there have been investigations into the analysis of filter-size variations. In this study, we aim to delve into this analysis, as it has served as the primary source of inspiration for the development of the Multi-Filter Convolutional Neural Network (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 show the impact of filter size on computational power and time complexity which also w. Ahmed and A. Karim [5] provide evidence of the impact of filter sizes on image sizes and highlights the exponential increase in time complexity associated with larger filter sizes in models. Furthermore, the findings demonstrate that, in certain instances, models employing bigger filter sizes yield comparable results to those utilizing 3x3 filter-based models. This observation aligns with the assertions made by Ozturk, U., Ozkaya, B., Akdemir, and L. [7], who suggest that both 3x3 and 7x7 convolution filters exhibit greater efficacy compared to 5x5 and 9x9 filters, as the former has the capacity to facilitate stronger learning processes rather than merely memorizing the data. these findings indicate that the utilization of large-scale filters in the context of the study has encountered the issues of overfitting as well as a significant rise in time complexity.

2.3 conclusion of review

The possibility arises that employing larger filters may confer certain advantages; however, such advantages may be rendered insignificant if outweighed by associated drawbacks. Two primary challenges, namely the exponential increase in time complexity and the risk of overfitting, act as significant barriers impeding researchers from unraveling the black box surrounding filter sizes. Consequently, investigating the potential benefits of larger filter sizes or devising solutions to address these challenges becomes essential to harness the advantages afforded by a potent learner.

3 Data and Methods

3.1 Dataset

The selection of datasets was conducted with meticulous consideration to leverage the advantages of the proposed layer while also uncovering potential drawbacks. To accomplish this, a set of predefined criteria and rules were established to identify specific datasets that align with the research objectives and enable a comprehensive exploration of the proposed layer's capabilities.

Rules

the proposed layer must be tested on big size dataset as well as medium dataset, we can exclude the small datasets because of the possibility of stronger learner overfit on such datasets. the number of classes

Flower Classification with TPUs

The Flower Classification with TPUs dataset holds a prominent position in the realm of computer vision research. It has emerged as a widely recognized and extensively utilized resource for training and assessing machine learning models designed specifically for classification tasks. This dataset offers a diverse assortment of high-resolution images depicting various species of flowers, thereby enabling researchers and data scientists to develop robust algorithms and test them. Originally hosted on the Kaggle platform, the dataset encompasses tens of thousands of color images capturing distinct flower species. These images exhibit variations in lighting conditions and backgrounds, thereby encompassing a comprehensive representation of real-world scenarios.A distinguishing feature of this dataset is the incorporation of Tensor Processing Units (TPUs) to facilitate accelerated computation. TPUs, specialized hardware developed by Google for machine learning tasks, bestow researchers with significant computational power, enabling expedited training of deep learning models.The Flower Classification with TPUs Kaggle dataset has gained considerable adoption among researchers and participants of Kaggle competitions, emerging as a benchmark for evaluating diverse algorithms and techniques in the domain of flower classification. Researchers leverage this dataset to explore an array of deep learning architectures, particularly convolutional neural networks (CNNs)[9].

Reason

- A semi-large-scale dataset is employed to demonstrate the impacts of the proposed layer.
- A diverse range of input image sizes is utilized to evaluate the efficacy of our proposed layer across three distinct image dimensions derived from the same dataset.
- A dataset encompassing numerous classes is utilized to illustrate the influence of the proposed layer on addressing the intricacies of complex globalization problems.

ISIC 2018 HAM10000

The HAM10000 dataset constitutes a valuable resource with significant implications for the advancement of dermatology and machine learning research. Comprising a vast assemblage of 10,015 high-resolution dermatoscopic images, it encompasses a diverse array of skin lesions that warrant meticulous analysis and scholarly investigation. The dataset's inherent strength lies in its scrupulous curation process, spearheaded by a cohort of esteemed dermatologists and researchers, ensuring meticulous annotation and validation to engender accurate and consistent labels for each lesion. Such painstaking attention to detail enhances the dataset's credibility and fosters confidence in its application for training and assessing machine learning algorithms. Incorporating both benign and malignant melanocytic and non-melanocytic lesions, the HAM10000 dataset affords researchers the opportunity to confront an assortment of diagnostic challenges, unravel intricacies, and explore intricate lesion characteristics. The inclusion of metadata, including demographic information such as age, sex, and lesion localization, augments the dataset's richness and facilitates investigations into potential associations between these variables and lesion features. The utilization of the HAM10000 dataset by researchers and data scientists has precipitated the development of state-of-the-art machine learning models for automated skin lesion classification, segmentation, and diagnosis[10].

Table 1: HAM10000 Data-set Table

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

Reason

- A dataset of moderate size is employed to observe the effects of the proposed layer on prevalent problems commonly encountered in the field.
- The presence of a substantial class imbalance prompts an investigation into the potential impact of the proposed layer on mitigating the effects of this challenging problem.
- A dataset consisting of seven classes exhibiting numerous Venn patterns is utilized to evaluate the efficacy of complex features that are recurrent across different classes.

SIIM-ISIC Melanoma Classification

The SIIM-ISIC Melanoma dataset is a highly valuable and widely employed compilation of dermatoscopic images exclusively focused on melanoma, the most lethal form of skin cancer. This dataset has emerged as a pivotal resource within the realm of computer-aided diagnosis and deep learning research in dermatology, serving as a critical foundation for the development and assessment of machine learning algorithms targeting enhanced early detection and diagnosis of melanoma. The SIIM-ISIC acronym denotes the collaborative efforts of the Society for Imaging Informatics in Medicine (SIIM) and the International Skin Imaging Collaboration (ISIC), two prominent entities dedicated to assembling an extensive and diverse collection of melanoma images. By amalgamating data from both organizations, the resultant dataset attains a comprehensive representation of melanoma cases. Researchers and data scientists harness the SIIM-ISIC Melanoma dataset to propel the development and evaluation of machine-learning models targeting melanoma detection, classification, segmentation, and risk assessment. Through leveraging this dataset, novel methodologies and algorithms can be explored, ultimately contributing to the automated identification and diagnosis of melanoma, thereby potentially enhancing patient outcomes and reducing mortality rates.[11]

Reason

- Assess whether our proposed layer is susceptible to challenges and limitations that arise specifically in binary classification scenarios.
- In a binary problem, The presence of a substantial class imbalance poses a significant challenge.

3.2 Multi-Filter CNN layer

The multi-filter layer encompasses a composition of several layers, each characterized by distinct filter sizes, resulting in output dimensions equivalent to those obtained from a standard CNN layer. The principle of the proposed layer resides in addressing potential challenges that arise from trying to create the multi-filter layer, thereby rendering it more applicable to real-time problems.

Output Dimension Challenge

The primary challenge encountered in the development of the multi-filter layer is the disparity in output dimensions that arise when utilizing different filter sizes.

$$out\ dimension = \frac{W + 2P - (K-1) - 1}{S} + 1 \quad (1)$$

The required solution needs an understanding of both convolution arithmetic and equation (1) [12] as the equation depends on four factors: input-size(W), filter-size(K), stride(S), and padding(P). Two factors are out of our control which are input and filter size so we need to modify the equation(1) to output the same dimension with different filter and input sizes. The other factors could be used to solve the dimension problem, we start by simplifying the equation by using the default stride which is one. this leaves us with padding, the only controllable factor which is by default is one.

$$P = \mathbf{ceil}(K/3) \quad (2)$$

we use padding to fill the gap between the output dimension of 3x3 kernels and higher kernels size, we calculate the needed padding for the size of each kernel manually and then created a simple equation (2) to automatically calculate the needed padding for any kernel size to output same dimension as 3x3 based filter layer.

$$out\ dimension = W + 2(\mathbf{ceil}(K/3)) - (K - 1) \quad (3)$$

we simplify the equation needed for our proposed layer in equation (3) that solves the dimension problem of using different filter sizes in the same layer so we could split the filter number on two or higher filter sizes and then simply concatenate the results of the different filter sizes.

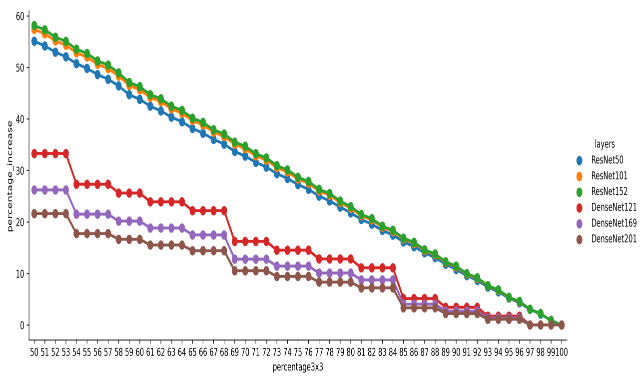

Figure 1: parameters increase percentage

Computational Power Challenge
the computational power needed to run bigger filters can reach an exponential rate due to the need to adjust many parameters. this exponential rate in computational power can prevent the proposed layer from being usable due to the time complexity. we calculate the percentage of trainable parameters that increase the more we lower the existence percentage of the 3x3 filter in the proposed layer. As Figure 1 shows that rate of increase in trainable parameters can reach 60% in the worst case this means that this model has 160% trainable parameters more than its original trainable parameters. Figure 1 also shows if we choose the right percentages, we can end up with a small increase in the trainable parameters.

Overfitting Challenge
The overfitting problem may occur due to the usage of stronger learners, we solve the stronger learner problem by making the 3x3 filter size percentage the dominant percentage to limit the number of bigger filters to limit memorizing training data while keeping the insights of stronger learners.

3.3 Multi-Filter CNN Structures

The Multi-filter layer is just a layer that can be used in any CNN structure to enhance the model by replacing the normal Convolution layer like in figure 2 and figure 3. There are

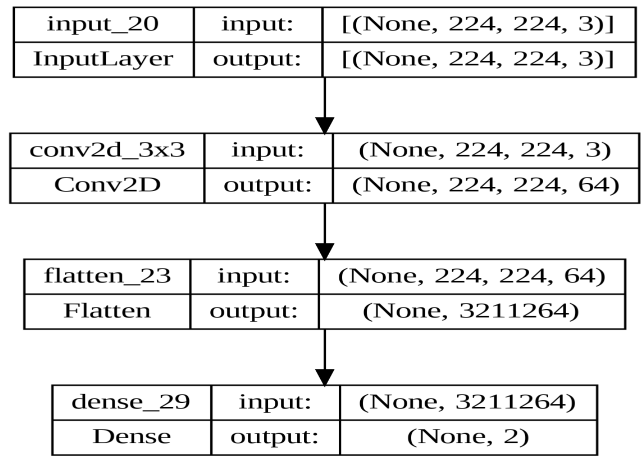

Figure 2: simple CNN Structure

many variants for the usage of the Multi-filter layer which can be only limited by machine learning engineers' imagination but we are going to show two simple structures due to the unlimited variants of the Multi-filter layer structures.

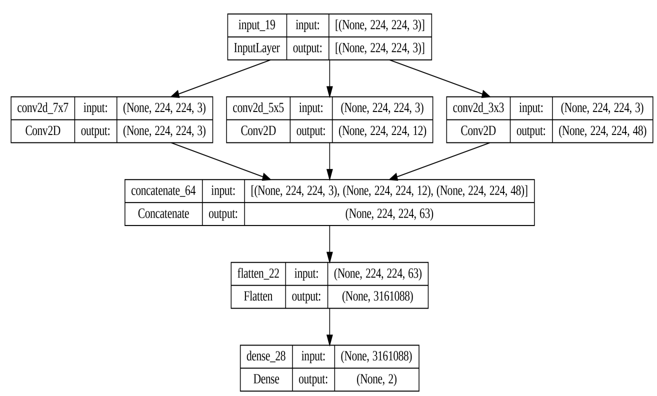

Figure 3: simple CNN Structure after proposed layer

Fixed Structure
The fixed structure is based on replacing each layer which is only based on a 3x3 filter size with a Multi-filter layer with a fixed percentage for each filter size that exists within the Multi-filter layer. Filter percentages do not change in all replaced layers. There are certain recommendations for choosing each filter size percentage of existence within The Multi-filter layer.

Recommendations (maybe in the discussion as revealed)

- Filter size 3x3 must exist with the chosen filter sizes (time complexity+ overfitting 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. (delete?)

Decreasing Structure (DS)

Decreasing structure is the same concept as the fixed structure yet instead of using a fixed percentage of filter size, we use decreasing equation which decreases the percentage of bigger filters based on the position of the 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 leads to lowering the trainable parameters (weights) that need to be adjusted. limiting overfitting by lowering bigger filter sizes percentage which lowers the existence of stronger learners in the upper layers of the structure that is responsible for feature extraction.

4 Results & Discussion

The proposed layer need to be exploited by a number of tests which should be efficient with tolerance for computational power resources available this is why we design a couple of tests to show both possible advantage and disadvantage of using the proposed layer. As any new approach tested need a big dataset to see if it can handle a big network of relations, the proposed layer is also used in feature extraction which rise the need of testing the effects of different image sizes. The depth of the model is a factor that affects the model performance so testing different depths with the proposed layer is also needed. Although we are limited by computational power to test all possible good percentages for fixed structure, we could test a different couple of filter sizes' percentages that show the effects of the proposed layer. those percentages are shown in Tables [2; 3; 4; 5] like 75-20-5 which represent filter-3x3-percentage & filter-5x5-percentage & filter-7x7-percentage in the same order if there is missing value, it means this filter size does not exist in this approach.

4.1 perentage effects

we started with two approaches for using three filters in the proposed layer and another two approaches for using only two filters in the proposed layer. The first approach (75-20-5) is the most promising in terms of results as it has the best mean approach in Tables [2; 3; 5] with an improvement that ranges approximately between 1% and 3%, it also has the best high improvement in a single model with improvement by 5%. The second approach (88-10-2) is less promising yet it gives an indicator of a chance of finding a better percentage by going lower than the first approach in terms of the percentage of 3x3 filter or in between. Although the bad performance of the second approach, the third approach (85-15) gets better results than the second approach which uses only two filters with near results to the first approach. The third approach gives an indicator that if there are better approaches, it must be by lowering the 3x3 filter percentage by more than 75% which has a risk of both time complexity and overfitting. The fourth approach(95-5) was tested to see the effects of a model with a small percentage of 5x5 filter size in it, the approach gets slightly better than normal but could not have better than the first approach yet there was an exception in table 4 where the image size was decreased.

4.2 image size

Tables [2; 3; 4] shows the impact of image size on the proposed layer, as it shows that the bigger the image the better for all the proposed layer approaches as not only the results decreases but also the improvement ranges decreases also which shows that the proposed layer perform better with bigger images which were expected due to using bigger filters sizes.Table 5 also shows using the proposed layer on small images could also have good results improvement yet having a bigger image is recommended.

4.3 depth

we used the three variants of both ResNet and Densenet to show the depth factor effects on the proposed layer. we can see an expected pattern for Densenet results which improves the bigger number of layers in Densenet yet in ResNet, we see the opposite pattern which can be explained by overfitting. overfitting is very visible in ResNet variants due to two reasons: using the proposed layer and image size.

4.4 best model

the best structure is Densenet in terms of results and stability in model mean in almost all results tables, yet one model variant of Densenet is almost the best model mean in all approaches which is Densenet201.

4.5 decreasing structure

5 Conclusion

6 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.

Table 2: 512x512 Flower Classification with TPUs

Models \ Approaches	normal	75-20-5	88-10-2	85-15	95-5	DS	Model Mean
Resnet50	0.702692	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 3: 331x331 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 4: 224x224 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 5: 224x224 ISIC 2018 Task 1

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

[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)*,

Table 6: 512x512 Flower Classification percentage difference

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 7: 331x331 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 8: 224x224 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 9: 224x224 ISIC 2018 Task 1 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

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] Tschandl and Philipp, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," 2018.

[11] A. Zawacki, B. Helba, G. Shih, J. Weber, J. Elliott, M. Combalia, N. Kurtansky, NoelCodella, P. Culliton, and V. Rotemberg, "Siim-isic melanoma classification," 2020. [Online]. Available: https://kaggle.com/competitions/siim-isic-melanoma-classification

[12] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2018.