A Pilot Study of Novel Multi-Filter CNN Layer

Mohamed Aboukhair[a], Abdelrahim Koura[a], and Mohammed Kayed[a]

[a] Computer Science Department, Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni Suef 62511, Egypt

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 used in any structure. Only at the first layers of the CNN model, filters bigger than 3x3 could be partially used. Most researchers work with filters (size, values, etc) as a black-box. To the best of our knowledge, this research is the first pilot study that proposes a new multi-filter layer in which different filters with variant sizes are used to replace the 3x3 filter layers. Our proposed multi-filter layer has yielded encouraging results, demonstrating notable improvements ranging from 1% to 5% in performance. This achievement was realized by developing two innovative structures, namely the Fixed structure and the Decreasing structure. Both of them leverage the Multi-filter layer. Although the two structures exhibit promising outcomes, the later structure offers the additional advantages of reduced computational requirements and enhanced learner strength.

Keywords: CNN, CNN structures, Classification

1 Introduction

A Convolutional Neural Network (CNN, or ConvNet), an Artificial Neural Network (ANNs), is the most commonly used model to analyze or classify visual imagery [1; 2]. A convolutional layer is the backbone of building CNN which extracts features on the basic and complex level of processing images inspired by the human brain [3; 4]. The convolutional layer consists of several filters, which can detect various features and their values are updated in the training stage. These filters are the most important part of CNN. Although, developers or researchers heavily depend on one size/dimension of the filter as a fixed size of all filters. It is not standard to use 3×3 filters, as researchers try to explore the effects of different filter sizes [5; 6; 7; 8]. It is only recommended to use a 3x3 filter size for three reasons: First, it has lower parameters to adjust in the training stage which speeds up the training process [5; 7]. Second, it does not support over-fitting because its size lowers the chance of memorizing the data [8]. Third, the need for a higher depth CNN model makes it harder to use bigger filters [5; 7]. 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.

Empirical evidence has substantiated the notion that several undesirable consequences emerge as filters expand in size within a neural network architecture. Basically, the learning process experiences a notable slowdown, impeding the overall efficiency of the model optimization. Additionally, there is a heightened susceptibility to overfitting, whereby the model becomes excessively tailored to the training data, impairing its generalization capabilities. Moreover, the overall complexity of the model amplifies as larger filter sizes necessitate a larger number of weights to be learned, potentially exacerbating computational demands and hindering scalability [9].

Given the substantiated facts surrounding these implications, researchers have gradually shifted their focus away from the effects of filter size variations. The observed drawbacks have instigated a reticence to explore extensive experiments involving diverse filter sizes. Instead, emphasis has been placed on alternative avenues for model optimization and performance improvement, such as architectural modifications, regularization techniques, or novel network structures [10; 11; 12]. Consequently, exploring filter size variations has become relatively limited, as the proven adverse effects outweigh the potential benefits in many research contexts.

To the best of our knowledge, no prior work has been found about using a single layer with multi-filter sizes to replace 3x3 filter size layer. Other researchers apply multiple filters separately with multiple layers to create the structure 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 is another limitation to explore the impact of applying bigger filters as it requires high computational power to overcome the time complexity[13].

This paper aims to open the black-box of the CNN layer by reviewing the effects of different filter sizes of related work and 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 that are higher in quality for CNN to enhance its performance. We avoid using too much bigger filters to prevent higher computational power and the need for more parameters to adjust with acceptable time to skip the exponential increase in time complexity [13].

The contributions of this paper are summarized as follows:

- 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

The remaining sections of this paper are organized as follows: Section 2 presents a comprehensive review of the previous literature relevant to our work. In Section 3, we discuss the datasets used in our study and introduce our novel Multi-Filter CNN layer along with the structures constructed based on it. Section 4 is dedicated to the presentation and analysis of experimental results, where we demonstrate the performance of the proposed Multi-Filter CNN layer and its structures on multiple datasets. Finally, in Section 5, we offer concluding remarks summarizing the key findings and contributions of this research.

2 Previous Literature

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.

Y. Camg ozl u and Y. Kutlu, as well as O. Khanday, S. Dadvandipour, and M. A. Lone, [5; 7] 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 [6] 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. [8], 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.

The Inception module, proposed in the seminal work ”Going Deeper with Convolutions,”[14] introduced a novel structure incorporating multiple filters. However, it should be clarified that the Inception module was not designed to function as a standalone layer or replace individual layers in a model. It faced challenges related to computational limitations[15] and managing output size growth associated with 5x5 Convolutional layers, which were addressed through input reduction strategies. Furthermore, the Inception module was recommended for utilization primarily in the upper layers of a model. Subsequent research, exemplified by ”Rethinking the Inception Architecture for Computer Vision,”[16] aimed to enhance the module by substituting 5x5 Convolutional layers with two consecutive 3x3 Convolutional layers. This modification aimed to improve efficiency and effectiveness while maintaining the overall module architecture. Nevertheless, it revealed that the original Inception module was not intended to incorporate filter sizes larger than 3x3. Further exploration and development of alternative approaches, such as the proposed multi-filter layer in this research, are crucial to address the limitations and challenges associated with the use of multiple filter sizes.

It should be noted that the proposed research introduces the multi-filter layer as a novel approach that surpasses the limitations and intended use of the Inception module, As 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 dataset selection was conducted meticulously to leverage the advantages of the proposed layer while also uncovering potential drawbacks. 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 datasets as well as medium datasets; we can exclude the small datasets because of the possibility of stronger learner overfit on such datasets.

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)[17; 18] 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)[19].

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[20]. 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[21].

Table 1: HAM10000 Data-set Table

Compare by Classes	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 [22].

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 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.

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

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

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

To address the issue of disparate output dimensions resulting from different filter and input sizes, a comprehensive understanding of convolution arithmetic and Equation (1) [23] is essential. Equation (1) relies on four factors: input size (W), filter size (K), stride (S), and padding (P). While input and filter sizes are beyond our control, modifications to the equation are necessary to ensure consistent output dimensions across the varying filter and input sizes. Consequently, the remaining factors, specifically stride, and padding, can be employed to tackle the dimension problem. Initially, Equation (1) is simplified by setting the stride value to one as the default, ensuring that smaller output sizes are not created due to bigger strides. This prevents excessive padding, which could result in the loss of information, as a result, the focus is placed on the controllable factor of padding. Padding is employed to bridge the discrepancy in output dimensions between 3x3 kernels and larger kernel sizes. To achieve this, a manual calculation is performed to determine the required padding for each kernel size. Subsequently, a straightforward equation, denoted as Equation (2), is devised to enable the automatic calculation of the necessary padding for any given kernel size, ensuring a consistent output dimension as that of a 3x3-based filter layer. Equation (3) is introduced as a simplification of the required equation for our proposed layer. This equation enables the partitioning of the filter count among two or more filter sizes, subsequently allowing for the straightforward concatenation of the resulting outputs from the distinct filter sizes.

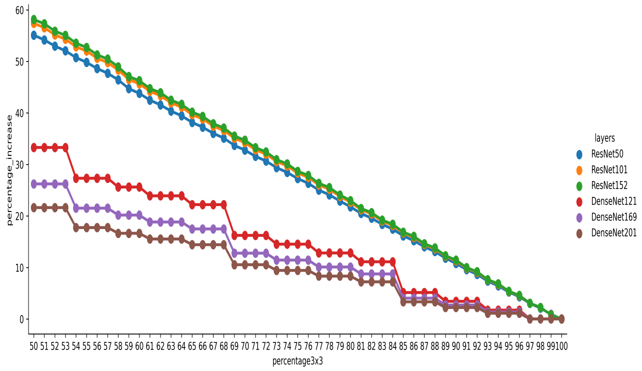

Figure 1: parameters increase percentage

Computational Power Challenge

The utilization of larger filters in the proposed layer can lead to a significant increase in computational power requirements, which exhibit an exponential growth pattern. This exponential increase poses a potential hindrance to the practical usability of the proposed layer due to the associated time complexity. The percentage of trainable parameters is calculated as standard, considering the reduction in the presence of the 3x3 filter within the proposed layer. Figure 1 illustrates that the increase in trainable parameters can reach up to 60% in the worst-case scenario, indicating that the model would possess 160% more trainable parameters than its original configuration. Figure 1 further illustrates that appropriately selecting the corresponding percentages makes it possible to achieve a marginal increase in the number of trainable parameters. For instance, employse allocation for the 3x3 filter size may result in a maximum utilization of approximately 130%, representing a worst-case scenario. This lower also the number of parameters needed to be trained, which skips the problem of time complexity that leads to bigger filters to get poor scores[9].

Overfitting Challenge

The issue of overfitting can arise when employing more potent learners.[24; 25] To mitigate this problem, we address the stronger learner dilemma by assigning a dominant percentage to the 3x3 filter size to lower model complexity. This strategy restricts the utilization of larger filters, thereby preventing excessive memorization of the training data while still benefiting from the insights provided by stronger learners.

3.3 Multi-Filter CNN Structures

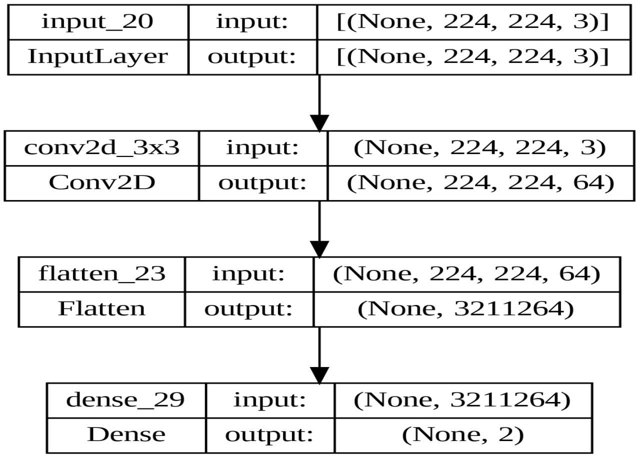

Figure 2: simple CNN Structure

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 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.

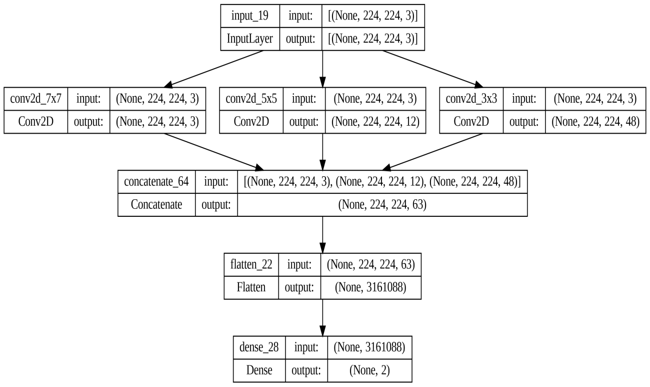

Figure 3: simple CNN Structure after proposed layer

Fixed Percentage Structure

The fixed structure employed in this approach involves substituting each layer, originally composed solely of a 3x3 filter size, with the Multi-filter layer that incorporates a consistent percentage allocation for each filter size present within the Multi-filter layer. Throughout the replaced layers, the distribution of filter percentages remains constant. As a result, this framework yields a more robust model that exhibits increased strength proportional to the increase of percentage allocation of larger filters.

Fixed Structure Percentage Choices

There are specific guidelines and recommendations pertaining to the selection of the percentage distribution for each filter size within the Multi-filter layer. These recommendations stem from the addressed challenges encountered during the enhancement of the Multi-filter layer, aiming to enhance its applicability in various contexts.

Recommendations

- Filter size 3x3 must exist with the chosen filter sizes
- Smaller filter size must have the dominating percentage over bigger filter size
- The bigger the filter size gets the smaller percentage assigned to it

By incorporating these recommendations and leveraging the insights gained from addressing the aforementioned challenges, we can proceed to assign specific percentages with a variant for experimental evaluation on the selected datasets.

Percentages:

- 75-20-5 and its variant 88-10-2 using 3 filters 3x3 , 5x5 and 7x7
- 85-15 and its variant 95-5 using 2 filters 3x3 and 5x5

Decreasing Structure (DS)

The concept of the decreasing structure shares similarities with the fixed structure approach, with the key distinction lying in the utilization of a decreasing equation(4) to determine the percentage allocation of larger filters based on the layer's position within the structure. Concurrently, the percentage assigned to the 3x3 filter size increases proportionally by the value derived from the larger filter sizes. As the layers progress from the top until reaching the flatten layer, the percentages gradually decrease, ultimately reaching zero for the larger filter sizes.

$$fx_i = int((F * P_{x_i}) * (1 - \frac{pos}{C})) \tag{4}$$

$$F_{3X3} = F - \sum_{Xi=1}^{n} fx_i \tag{5}$$

Equation (4) delineates the decreasing equation utilized to determine the filter count for larger filter sizes within the Multi-filter layer. Here, fx_i denotes the filter count for a specific filter size (x_i), while F represents the total allowable count of filters in the layer. P_{x_i} represents The probability corresponding to the likelihood of a particular filter size being present in the Multi-filter layer. Additionally, pos signifies the position of the layer within the list of replaced layers, and C denotes the total count of all replaced layers. The second component of Equation (4) serves the purpose of inversely adjusting the probability of filter existence, facilitating an increase in the representation of larger filter sizes at the upper layers while progressively reducing their prevalence towards the lower layers. This strategic manipulation effectively curtails the dominance of stronger learners in the network architecture, striking a balance that mitigates the risk of computational overhead while still leveraging the benefits offered by stronger learners.

Equation (4) exclusively computes the filter count for larger filter sizes, neglecting the primary filter size of 3x3. In contrast, Equation (5) determines the number of filters allocated to the 3x3 filter size by aggregating the filter counts of the larger filter sizes. This approach ensures the comprehensive consideration of both the larger filter sizes and the fundamental 3x3 filter size within the calculation of the filter distribution.

DS Percentage Choices

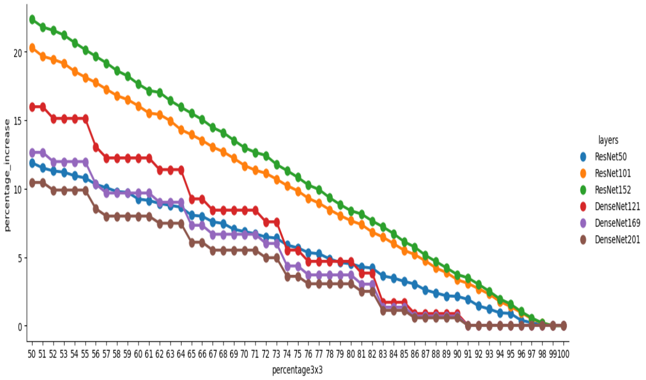

Figure 4: parameters increase percentage DS

The recommendations for the Decreasing Structure approach closely resemble those of the Fixed Structure. However, Figure 4 demonstrates that the Decreasing Structure offers greater flexibility in reducing the dominance of smaller filter sizes. This adjustment creates opportunities for incorporating stronger learners while simultaneously lowering the time complexity by reducing the percentage allocation of larger filter sizes in the lower layers. Consequently, this reduction in the percentage of larger filters contributes to a decrease in the number of trainable parameters, thereby diminishing the weight adjustment requirements. Furthermore, the limitation of overfitting is achieved by reducing the presence of stronger learners in the lower layers of the structure and increasing them in the upper layers, which play a pivotal role in feature extraction. Taking the aforementioned considerations into account, it is possible to explore a configuration where the dominant percentage is set at 60% for all filter sizes within the proposed layer. This implies that the first filter size would be allocated 60% of the available filters, while the subsequent filter sizes would each receive 60% of the remaining filters after the allocation to the previous filter size and etc.

4 Results & Discussion

To effectively evaluate the proposed layer, a series of tests are required, taking into account the available computational resources and ensuring computational efficiency. As is customary when examining novel approaches, a sizable dataset is necessary to assess the capability of the proposed layer in handling complex relational networks. Furthermore, given that the proposed layer is primarily utilized for feature extraction, it is imperative to investigate the impact of different image sizes on its performance. Additionally, the depth of the model is a crucial factor influencing overall performance, necessitating the testing of various depths in conjunction with the proposed layer.

Due to computational limitations, conducting exhaustive tests encompassing all potential configurations of the fixed structure may not be feasible. However, it is possible to conduct experiments with selected filter size percentages, representing different combinations such as 75-20-5, as shown in Tables [2; 3; 4; 5]. In this representation, the percentages denote the allocation of filters in the order of filter-3x3-percentage, filter-5x5-percentage, and filter-7x7-percentage. Any missing value in the sequence indicates that the corresponding filter size is not employed in the proposed layer for that particular configuration. These test results are typically presented in tabular form, allowing for a comprehensive analysis of the proposed layer's effects and implications.

4.1 Percentage Effects

Initially, two distinct approaches were employed to utilize the proposed layer, each involving three filters. Additionally, two alternative methods were investigated, using only two filters within the proposed layer. The first approach, denoted as (75-20-5), exhibited the most promising outcomes based on the mean improvement observed across Tables [2; 3; 5]. The results indicated an improvement ranging from approximately 1% to 3%, with a notable high improvement of 5% in a single model. In contrast, the second approach, represented as (88-10-2), demonstrated less promising results, suggesting the potential for further exploration by either lowering the percentage of the 3x3 filter or considering alternative percentages between the filters. Despite its relatively poor performance, the third approach (85-15), which employed only two filters, displayed improved results compared to the second approach, exhibiting outcomes that closely approximated those of the first. The exploration of a fourth approach (95-5) aimed to assess the effects of incorporating a small percentage of the

Table 2: 512x512 Flower Classification with TPUs

Approaches Models	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

Approaches Models	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

Approaches Models	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

Approaches Models	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

5x5 filter size in the model. Although this approach yielded slight improvements compared to the baseline, it failed to surpass the performance achieved by the first approach. However, an exception was noted in Table 4, where a reduction in image size was implemented. In summary, the results from these diverse approaches provide insights into the potential for optimizing the performance of the proposed layer. Lowering the percentage allocation for the 3x3 filter size may present opportunities for further improvements; however, it necessitates careful consideration of potential risks associated with increased time complexity and overfitting.

4.2 Image Size

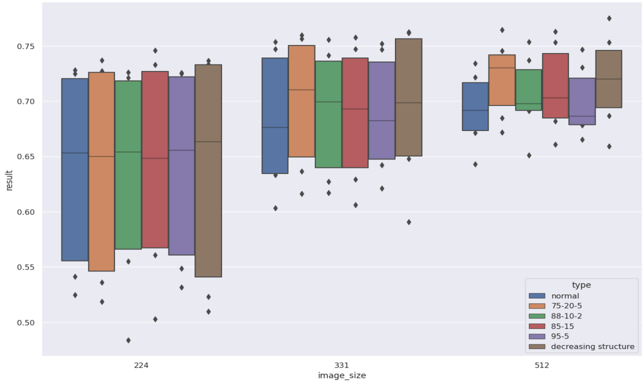

Figure 5: DS in Flower Classification with TPUs

The findings presented in Tables [2; 3; 4] and Figure 5 provide insights into the relationship between image size and the performance of the proposed layer. The results demonstrate that larger images yield superior outcomes across all approaches employed within the proposed layer. The increased results support this observation. It is, therefore, evident that the proposed layer exhibits improved performance when applied to larger images, which aligns with the initial expectations given the utilization of larger filter sizes. Additionally, Table 5 presents an interesting finding regarding applying the proposed layer to small images. Despite the reduced image size, notable performance improvements were still observed. However, it is essential to note that while the proposed layer can yield favorable results with smaller images, utilizing larger images whenever possible is recommended to maximize the potential benefits and enhancements provided by the proposed layer.

4.3 Depth

To examine the impact of the depth factor on the performance of the proposed layer, three variations of both ResNet and Densenet architectures were employed. The findings revealed an anticipated pattern in the results obtained from Densenet, wherein increasing the number of layers led to improved performance. This pattern aligns with the inherent characteristics of Densenet, which fosters feature reuse and information flow across layers [26].

In contrast, a contrasting pattern emerged in the results obtained from ResNet variants. Here, an inverse relationship between the number of layers and performance was observed. This phenomenon can be attributed to the presence of overfitting, which was particularly evident in ResNet models. Two key factors contribute to the observed overfitting: the usage of the proposed layer and the image size employed.

The utilization of the proposed layer introduces a level of complexity to the ResNet architecture, which may exacerbate overfitting tendencies. Furthermore, the impact of image size cannot be overlooked, as larger images provide a more diverse and representative dataset, thereby reducing the likelihood of overfitting. These combined factors contribute to the visible overfitting effects observed in the ResNet variants, highlighting the importance of carefully managing model depth and considering the potential challenges associated with overfitting when applying the proposed layer within the ResNet architecture.

4.4 Best Model

Among the evaluated model architectures, Densenet demonstrated superior performance and stability in terms of results across various evaluation metrics. Specifically, the Densenet structure consistently achieved the highest mean performance across multiple results tables, indicating its favorable suitability for the proposed layer. Notably, a specific variant of Densenet, namely Densenet201, exhibited exceptional performance and consistently outperformed other models regarding mean results across all approaches. These findings underscore the preference for Densenet as the optimal architecture choice when integrating the proposed layer. The observed advantages of Densenet in terms of broader improvement ranges, stability, and consistently high model mean further validate its compatibility with the proposed layer. Such insights emphasize the potential benefits of leveraging the Densenet structure in conjunction with the proposed layer to enhance performance and stability in various deep-learning tasks

4.5 Decreasing Structure

Figure 5 depicts the comparative analysis of the stability and performance outcomes between the Decreasing Structure and the 75-20-5 Fixed structure variant. It is evident from the figure that both structures exhibit similarities in terms of stability and improved results. However, the Decreasing Structure surpasses the Fixed Structure by outperforming it in terms of the best model and showing a lower parameter increase. The Decreasing Structure offers superior results with minimal effort, as evidenced by its significantly lower parameter increase than the Fixed Structure. Furthermore, the Decreasing Structure enhances learner strength, surpassing the Fixed Structure's limitations. This advancement opens up possibilities for exploring diverse and creative structures utilizing the Multi-filter layer, which can potentially exceed the performance of the Decreasing Structure.

4.6 SIIM-ISIC Melanoma Classification

The results for the SIIM-ISIC Melanoma Classification dataset were not included in the analysis presented in this study. This omission was due to the findings indicating that the Multi-filter layer may not be suitable for binary classification problems. Specifically, the scores obtained from the Multi-filter layer showed minimal differences, typically below half a percent, or tended towards zero or near-zero values in both positive and negative directions. These results suggest that the application of the Multi-filter layer is not recommended for binary classification tasks.

5 Conclusion

In conclusion, the proposed multi-filter layer has demonstrated promising performance improvement across various approaches. The Fixed Structure and Decreasing Structure variants of the proposed layer have shown the potential to enhance model outcomes. The Decreasing Structure, in particular, offers the advantage of achieving better results with lower computational power requirements and increasing learner strength. Comprehensive tests were conducted to evaluate the proposed layer, considering the available computational resources and efficiency. Examining large datasets and analyzing different image sizes have provided valuable insights into the layer's capabilities. Larger images have consistently yielded better results, indicating the benefits of utilizing the proposed layer with larger filter sizes. The impact of the nature of the classification task is also essential as the multi-filter layer tends to perform well on multi-class problems and gives no difference from the typical approach on binary classification problems. The impact of model depth on the proposed layer's performance was also investigated, with Densenet architecture showcasing superior results and stability compared to ResNet variants. Densenet201 emerged as the most effective model variant across all approaches, highlighting its compatibility with the proposed layer and its potential for achieving high-performance outcomes. Furthermore, the comparison between the Decreasing Structure and Fixed Structure indicated the former's superiority, exhibiting better model performance and a lower parameter increase. The Decreasing Structure offers a promising avenue for further exploration and the development of innovative structures leveraging the Multi-filter layer. In summary, the proposed multi-filter layer has shown significant potential in enhancing model performance. The findings underscore the importance of carefully selecting the structure, percentage of the different filters, image size, and model depth when integrating the proposed layer into deep-learning tasks

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] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," Aug. 2017.

[3] 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.

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

[5] 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

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

[7] 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.

[8] 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

[9] O. Khanday and S. Dadvandipour, "Convolution neural networks and impact of filter sizes on image classification," *Multidiszciplináris Tudományok*, vol. 10, pp. 55–60, 04 2020.

[10] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, no. 8, pp. 5455–5516, Apr. 2020. [Online]. Available: https://doi.org/10.1007/s10462-020-09825-6

[11] M. A. Saleem, N. Senan, F. Wahid, M. Aamir, A. Samad, and M. Khan, "Comparative analysis of recent architecture of convolutional neural network," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–9, Mar. 2022. [Online]. Available: https://doi.org/10.1155/2022/7313612

[12] I. Rodriguez-Conde, C. Campos, and F. Fdez-Riverola, "Optimized convolutional neural network architectures for efficient on-device vision-based object detection," *Neural Computing and Applications*, vol. 34, no. 13, pp. 10 469–10 501, Dec. 2021. [Online]. Available: https://doi.org/10.1007/s00521-021-06830-w

[13] 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.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[17] N. P. Jouppi, C. Young, N. Patil, and Patterson, "In-datacenter performance analysis of a tensor processing unit," 2017. [Online]. Available: https://arxiv.org/abs/1704.04760

[18] D. Paper, "Introduction to tensor processing units," in *State-of-the-Art Deep Learning Models in TensorFlow*. Apress, 2021, pp. 127–152. [Online]. Available: https://doi.org/10.1007/978-1-4842-7341-8_5

[19] 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

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

[21] M. A. A. Milton, "Automated skin lesion classification using ensemble of deep neural networks in ISIC 2018: Skin lesion analysis towards melanoma detection challenge," *CoRR*, vol. abs/1901.10802, 2019.

[22] 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

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

[24] O. A. Montesinos López, A. Montesinos López, and J. Crossa, "Overfitting, model tuning, and evaluation of prediction performance." Cham: Springer International Publishing, 2022, pp. 109–139.

[25] M. Valdenegro-Toro and M. Sabatelli, "Machine learning students overfit to overfitting," 2022.

[26] A. Hess, "Exploring feature reuse in densenet architectures," *CoRR*, vol. abs/1806.01935, 2018. [Online]. Available: http://arxiv.org/abs/1806.01935