IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 April 2023, accepted 19 May 2023, date of publication 26 May 2023, date of current version 1 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280191

RESEARCH ARTICLE

Evaluating the Potential of Wavelet Pooling on Improving the Data Efficiency of Light-Weight CNNs

SHIMAA EL-BANA[1], (Student Member, IEEE), AHMAD AL-KABBANY[2,3], (Member, IEEE), HASSAN M. ELRAGAL[1], (Senior Member, IEEE), AND SAID EL-KHAMY[1], (Life Fellow, IEEE)

[1]Department of Electrical Engineering, Alexandria University, Alexandria 21544, Egypt
[2]Intelligent Systems Laboratory, Arab Academy for Science, Technology, and Maritime Transport, Alexandria 21937, Egypt
[3]Department of Research and Development, VRapeutic Inc., Cairo 11613, Egypt

Corresponding author: Ahmad Al-Kabbany (alkabbany@ieee.org)

ABSTRACT Wavelet pooling (WP) in neural network architectures has recently demonstrated more discriminative power than traditional pooling methods. This is mainly because the latter suffer from spatial information loss while wavelet pooling harnesses the power of spectral information. However, the potential of WP in increasing the data efficiency and the extent of this potential have not been investigated yet. Data efficiency refers to the volume of training data required to attain a certain performance level during inference, e.g., recognition accuracy. In this research, we are concerned with evaluating the data efficiency of WP in light-weight architectures–MobileNets. Across a wide variety of seven datasets/applications including object recognition (CIFAR-10, STL-10, CINIC-10, and Intel Image Classification datasets) and diagnostic imaging (colon diseases, brain tumors, and malaria cell images datasets), and while considering classification accuracy as a performance metric, we show that WP achieves an average data saving that exceeds 30% compared to traditional pooling techniques. For other performance measures, namely, precision, recall, and F1-score, we report an average of 30% data saving for object recognition datasets and 22% saving for diagnostic imaging datasets. By focusing on a light-weight architecture, this research further emphasizes the significance of wavelet pooling in training and testing resources-challenged settings such as the applications of edge computing and green deep learning.

INDEX TERMS CNNs, classification, MobileNet, pooling, wavelet, spectral information.

I. INTRODUCTION

Deep learning has defined the state-of-the-art (SOTA) for over a decade in various applications that involve signal processing, analysis, synthesis, and communication. Nevertheless, the computational efficiency for training and testing deep models have posed several challenges in use cases. For example, many problems that demand high level of privacy and security, training *data scarcity* has represented a significant obstacle towards harnessing the power of deep models. Data augmentation and transfer learning have helped closing the performance gap to some extent, but there is still room for further improvement. In other applications, data is not scarce, but the application framework would dictate certain distribution of computational workload. For example, in *edge computing* [1], [2], the distribution of the computational workload between the cloud and the edge devices (that are closer to the end user) would hinder system architects from harnessing the recognition power of large deep models due to the constrained resources of edge devices. Other areas that would benefit from deep learning might not suffer from data scarcity and might not involve edge computing, but reducing the systems' carbon footprint and energy usage, during model training and testing, is one of their ultimate goals–*green deep learning* [3]. The aforementioned contexts, i.e., data scarcity, edge computing, and green deep learning share the requirement of efficient usage of data.

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Kamrul Hasan.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Towards the efficient usage of data, the research literature on reducing the volume of training datasets has featured diverse research directions. One approach is known as data distillation [4], and it aims to create a dataset that is smaller than the original dataset. The synthesis process is guided by the resulting accuracy level, i.e., the accuracy attained using the model trained on the distilled dataset should match that of the model trained on the complete dataset. Another approach adopts a clustering framework [5] where samples from the complete dataset are first aggregated using the K-means algorithm or the k-nearest neighbour algorithm, then the samples that fall within the same cluster are combined. Data pruning is another direction that considers a model's ability to generalize well as a guiding metric to eliminate specific data samples from the training dataset [6]. Previous research had also investigated the direction of sample ranking. In one of these approaches, the samples of the training dataset are first ranked according to their discriminative power then the lowest ranked samples are removed in an iterative manner [7].

In this research, we investigate the potential of an emerging pooling technique, namely, the discrete wavelet transform-based pooling (WP), in improving the training data efficiency of light-weight neural network architectures. We define the training data efficiency as the amount of training data that a learning model needs to achieve a certain performance level. By considering the information in the different sub-bands of the wavelet domain, WP widens the receptive field, similar to other pooling techniques, while retaining spectral information that are usually lost in average and max pooling. Wavelet pooling was first introduced in the literature to overcome spatial information loss which is an inherent drawback in traditional pooling methods. It has been implemented with various deep architectures, in classification and segmentation applications, and solely as well as in hybrid pipelines with traditional pooling methods [8], [9], [10]. Also, the incorporation of all sub-bands, approximation sub-band only, as well as matching the input images to a specific combination of wavelet sub-bands [11] has been investigated in the literature.

Previous research on wavelet pooling has not investigated its potential with regards to data efficiency. In [12] and [13], we highlighted the capacity of WP in improving the recognition accuracy of lightweight deep models with a focus on objects and remote sensing datasets. In [11], we showed, using extensive simulations, that the best recognition accuracy does not necessarily coincide with including the four sub-bands of the first level wavelet decomposition, and that choosing the sub-bands to include in an adaptive manner guarantees higher recognition accuracies. Furthermore, other previous studies on WP had shown that the accuracy level attained using WP is comparable or can eclipse, i.e., is within respectable ranges or higher, compared to the level attained using other traditional pooling techniques, keeping other training parameters the same in the models being compared [12], [14], [15]. However, the following research question has not been addressed yet: *Given that model X adopts WP and model Y adopts traditional pooling, how much training data can be discarded, while training model X, until the model performance degrades to that of model Y?*

We consider one flavor of wavelet pooling-base MobileNet in this research. This model considers all the sub-bands in a first-level wavelet decomposition [12]. On seven object detection and diagnostic imaging datasets that are shown in Fig. 1, we report an average training data saving that exceeds 30%, taking the classification accuracy as our performance measure. For the precision, recall, and F1-score, we obtained an average data reduction that exceeds 30% and 22% for object recognition datasets and diagnostic imaging datasets, respectively. By focusing on a light-weight architecture, which is specially useful in contexts where testing resources are scarce, we aim at highlighting the potential of WP when the training resources are also scarce.

The rest of this article is organized as follows. The second section highlights the recent literature on data-efficient models and wavelet pooling-based models. Section III is devoted to discussing the details of the proposed pipeline. Section IV presents the results obtained using the proposed model before section VI concludes the research and identifies relevant future research directions.

II. RELATED WORK

This section is dedicated to discussing the research literature that overalps with the scope of this article. Particularly, we highlight the recently proposed methods that aimed at reducing the volume of the training datasets without compromising the model's performance. Furthermore, we cover the recent literature on wavelet transform-based pooling in CNN models, and we identify the two variants that will be adopted in the rest of this research.

A. LITERATURE ON DATA EFFICIENCY

The development of model learning pipelines that get trained on a subset of the training dataset, without compromising the models' performance, have recently attracted an increasing attention from the research community. Towards this goal, one approach is to use data distillation rather than retaining a subset of the entire dataset [4]. By distillation, they mean the synthesis process of a smaller dataset that would not degrade the accuracy level of the model under consideration. Over several training iterations, the authors proposed to optimize the distilled data such that they minimize the distance between the parameters of the network being trained and the parameters of the network trained on the entire dataset. Using this formulation, they guide the network that is being trained to a similar state as a network trained on the whole dataset across many training steps while enhancing data distillation.

Dataset pruning is another framework that was proposed to downsize a training dataset by examining the influence of eliminating specific sets from the training data on the generalization accuracy of the model [6]. Then, the authors proposed to generate the smallest amount of training data

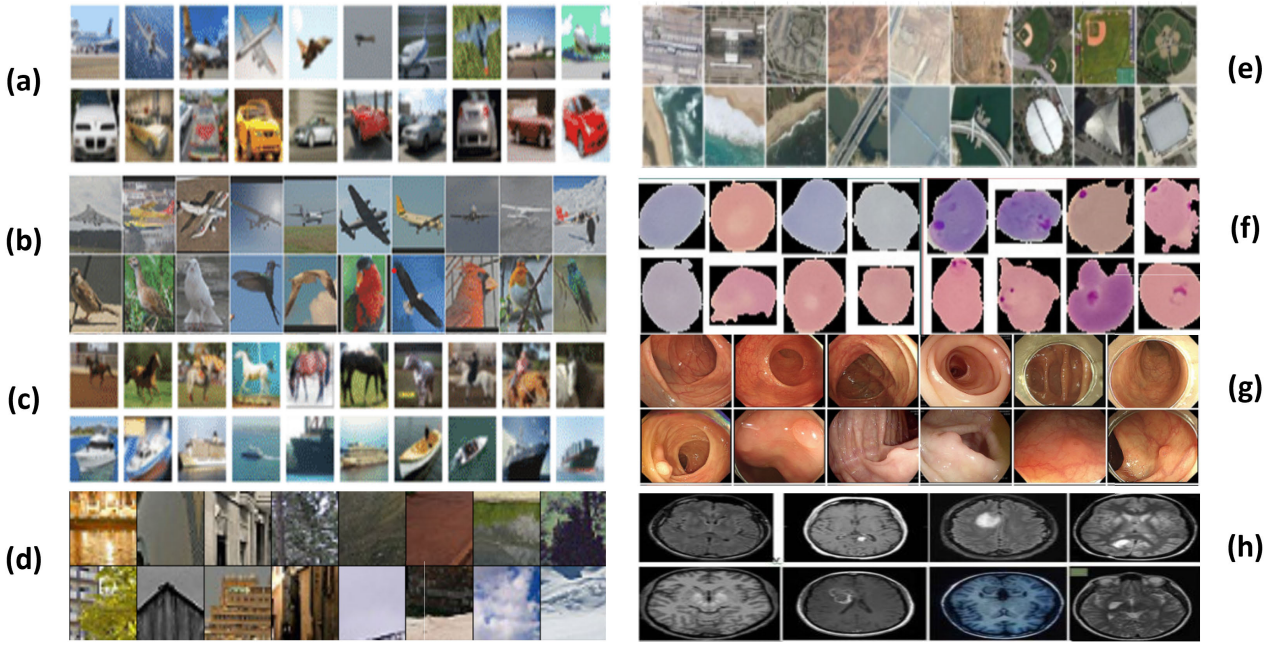

FIGURE 1. Samples from the diverse datasets adopted in our research. The applications of these datasets include object recognition such as: a) CIFAR-10, b) STL-10, C) CINIC-10, and d) Intel Dataset, remote sensing such as e) Land Scene remote sensing images, and diagnostic imaging such as f) Malaria Cell dataset, g) WCE Curated Colon Disease dataset, and h) Brain Tumor diseases.

that produce a highly restricted generalization gap. Using this framework, the solutions obtained through training are required to be competitive with those obtained in the full dataset.

Clustering-based approaches starts by using widely used techniques such as k-means [5] and k-nearest neighbor [16] to aggregate dataset samples into clusters. Afterwards, the dataset size can be decreased by combining samples from the same cluster. The complexity of the k-nearest neighbor algorithm and k-means algorithms, on the other hand, make this approach difficult to adopt with small datasets. Using coresets, which are weighted subsets of the entire dataset, is another strategy for reducing the size of the training set of data [17].

In order to reduce the number of training samples, Benyamin et al. [7] proposed a technique known as Principal Sample Analysis (PSA). In PSA, each sample in the set is ranked according to how well it could be utilised to distinguish between different data classes. The PSA algorithm removes the samples with the lowest rankings in an iterative manner. Moreover, the authors of [18] proposed a formula for estimating an example's training value and employ it for ranking other examples greedily. Nevertheless, sample ranking approaches, so far, fit exclusively with classification tasks. Also, they do not scale well with big datasets.

B. WAVELET POOLING

In general, pooling methods like stride, average and max pooling are frequently employed for down-sampling operations and expanding the receptive field such as in [19], [20], [21], and [22]. Nevertheless, they might cause a significant loss of information. In order to overcome this drawback, recent literature had featured several investigations to combine the discrete wavelet transform as a pooling method with deep learning architectures. For instance, Juan et al. [8] provided a classification system that adopts WP. Particularly, they use the first level decomposition of an image. This level contains 4 sub-bands that contain the approximation, horizontal, vertical, and diagonal features. Beside classification, AndréB) et al. [9] proposed to use WP in semantic segmentation. They presented a hybrid pooling scheme which combines wavelet and traditional pooling. This strategy was applied on a new version of the Segnet model–MPSegnet.

In order to balance the size of the receptive area and the computational performance, the multi-layer wavelet CNN (MWCNN) approach was proposed [14]. The incorporation of wavelet transform within the CNN architecture was shown to minimize feature maps. Moreover, the MWCNN technique was also applied on a U-Net architecture and the inverse wavelet transform (IWT) for high-resolution image restoration. Moreover, Qiufu Li et al. [10] proposed a 3D wavelet-based neuron segmentation method–3D WaveUNet. Their proposed model was proven effective to deal with the fine structured neurons that spread over a large area while avoiding the high computational cost that is inherent to the segmentation process of such structures. That model was also shown to effectively handle damaged fibers and the high levels of noise that characterize this problem.

In our previous work, we implemented various flavors of wavelet pooling strategies on light-weight models [12], particularly with MobileNets [23], which are based on depth-wise separable convolution and factorized Networks [24].

MobileNet's primary layer involves depthwise separable convolution. This is a sort of factorised convolution that splits the standard convolution into a depthwise convolution and a 1×1 convolution. We applied WP on various datasets, including multi-label remote sensing tasks [13] to harness the power of spectral information in deep networks. Furthermore, We present a new training and inference approach called Matched Wavelet Pooling (MWP) that determines which sub-bands should be used in the pooling operation for every image during training and testing [11]. This was based on the observation that including all of the wavelet decomposition's sub-bands does not always result in higher performance than utilising a specific subset of sub-bands. In this research, we present results for the wavelet-based pooling that considers all the sub-bands in the first-level wavelet decomposition of an image.

III. PROPOSED METHOD

In sec. I, in order to motivate the proposed research, we stated three examples for scenarios that require efficient usage of data, namely, data scarcity, edge computing, and green deep learning. This section is dedicated to discussing the proposed methods for evaluating the data efficiency of wavelet-based pooling implemented in light-weight deep architectures, namely, MobileNets (**WaveMobileNets**). By focusing on a light-weight architecture, which is specially useful in contexts where testing resources are scarce, we aim at highlighting the potential of WP when the training resources are also scarce. We start off by discussing the implementation of WaveMobileNets. The weights of the wavelet filters and the type of the mother function used in our experiments will be given in this discussion. We will also present the different layers of the proposed MobileNet-based model, highlighting the layers that involve wavelet pooling. Afterwards, we explain our approach for simulating different levels of data availability, since a good model's performance with a small volume of available training data would imply high data efficiency. The discussion on varying data availability will also involve a justification for key design decisions in our experiments including the range of dataset's size variation. Lastly, we conclude the section by presenting the details of the datasets adopted in our experiments.

A. DISCRETE WAVELET POOLING IN MobileNets

Following our previous work [12], we employ a modified version of MobileNet architecture that uses DWT pooling layers, which was shown to achieve significantly better results compared to standard MobileNet in [12] and [23]. Figure 2 depicts the architecture of our model and the various training datasets fed to it. These training datasets represent sub-sampled versions of the whole training dataset which is initially constructed using an 80-20 train-test splitting. We elaborate on the construction of these datasets in the rest of this section. The model is shown to comprise five cascaded stages, where the construction of every stage is indicated by the legend at the bottom of the figure. The second and the fourth stages feature wavelet pooling layers which are computed as explained in the following lines. In our model, a convolution block is comprised of a depth-wise convolution (DW), a point-wise convolution (PW), and a depth-wise separable convolution (DWCNN). The construction of a DWT block is similar to that of a convolution block except that a DWT pooling layer is inserted between the DW and the PW layers.

Given an image G of size (n, n, m), we propose to construct the pooling layer using the Haar basis function which is formed by the two filters $l = (1/\sqrt{2}, 1/\sqrt{2})$ and $h = (1/\sqrt{2}, -1/\sqrt{2})$. In the output of the first-level 2D DWT, G_{LL} represents the low-frequency sub-band of the input G, which usually contains most of the information in natural images. The other structures which are G_{HL}, G_{LH}, and G_{HH} represent components of high-frequency which contain the vertical, horizontal, and diagonal details of G, respectively. The four filters of Haar wavelets have fixed parameters with convolutional stride 2 during the transformation. These filters are defined as:

$$K_{LL} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \; K_{LH} = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}, \\ K_{HL} = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \; K_{HH} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}. \tag{1}$$

In this research, we propose to realize filtering and downsampling in MobileNet-V1 as depthwise separable convolution (DWCNN in Fig. 2 and DWT pooling. This formulation facilitates connecting MobileNets with multi-level wavelet analysis. For computing the 4 subband images G_{LL}, G_{LH}, G_{HL}, and G_{HH}, the mathematical expressions are given as:

$$G_{LL} = (K_{LL} \otimes G) \downarrow 2; \quad G_{LH} = (K_{LH} \otimes G) \downarrow 2 \\ G_{HL} = (K_{HL} \otimes G) \downarrow 2; \quad G_{HH} = (K_{HH} \otimes G) \downarrow 2, \tag{2}$$

where stride $\downarrow 2$ denotes the downsampling operator with factor 2, and $\otimes$ denotes the convolution operator. Furthermore, based on the principle of the Haar transform, the (q, r)-th value of G_{LL}, G_{LH}, G_{HL} and G_{HH} after a 2D Haar transform can be represented as:

$$\begin{aligned} G_{LL}(q,r) &= G(2q-1,2r-1) + G(2q-1,2r) \\ &\quad + G(2q,2r-1) + G(2q,2r) \\ G_{LH}(q,r) &= G(2q-1,2r-1) - G(2q-1,2r) \\ &\quad + G(2q,2r-1) + G(2q,2r) \\ G_{HL}(q,r) &= G(2q-1,2r-1) + G(2q-1,2r) \\ &\quad - G(2q,2r-1) + G(2q,2r) \\ G_{HH}(q,r) &= G(2q-1,2r-1) - G(2q-1,2r) \\ &\quad - G(2q,2r-1) + G(2q,2r). \end{aligned} \tag{3}$$

The proposed WaveMobileNets is based on MobileNet-V1. Specifically, we propose to replace MobileNet-V1's

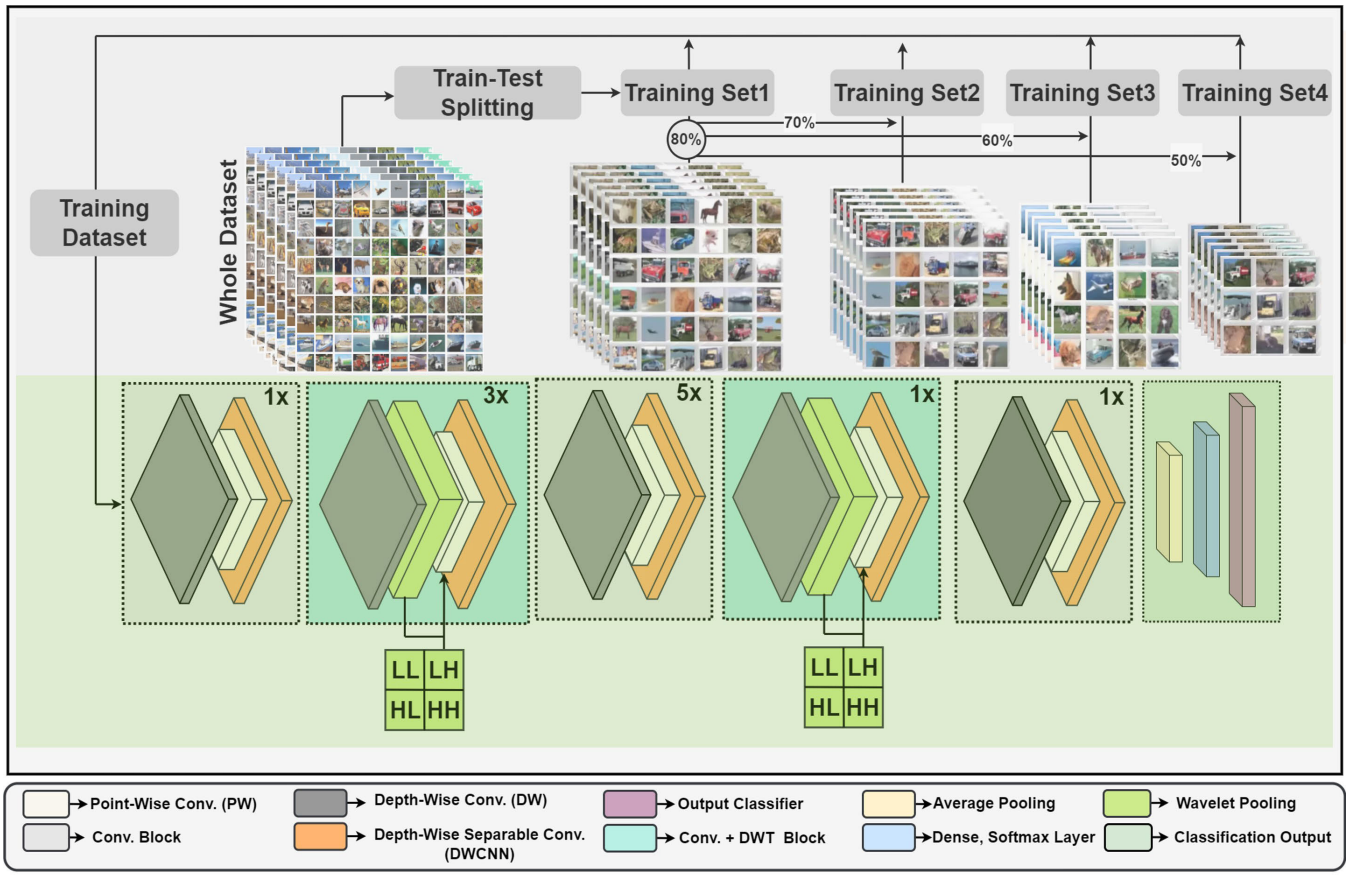

FIGURE 2. A depiction of the whole proposed framework. In the upper part of the figure, we show the dataset management/handling component which shows how different training datasets are constructed. The lower component (with highlighted background) shows the deep lightweight model to which the training datasets are applied.

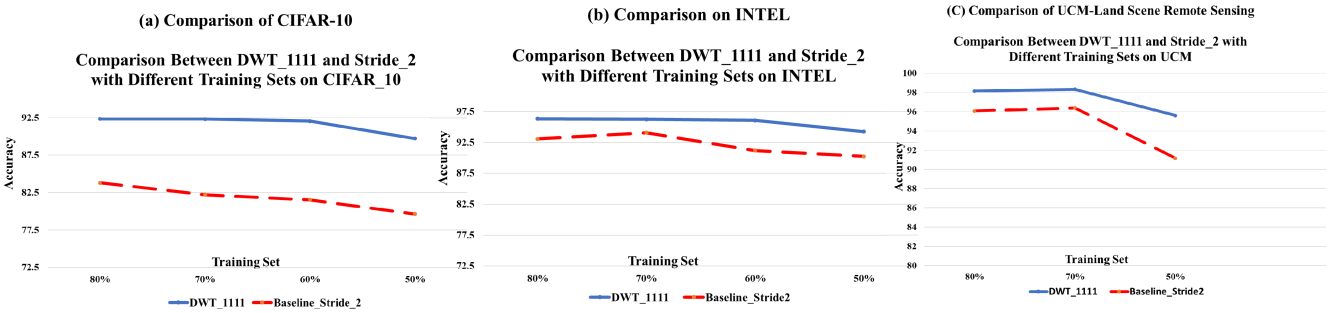

FIGURE 3. Justifying the chosen checkpoints (investigation range) at which the performance of the models under consideration (the WP-based MobileNet and the baseline MobileNet) is compared. Please see the text for more details.

down-sampling operations with $DWT_{LL,LH,HL,HH}$, i.e.,

$$\begin{cases} MeanPooling_{\downarrow 2} \\ MaxPooling_{\downarrow 2} \\ Stride_{\downarrow 2} \end{cases} \xrightarrow{\text{Wavelet Transform}} \begin{cases} DWT_{(LL,LH,HL,HH)}. \end{cases} \tag{4}$$

The detailed description of the WaveMobileNets [12] is as follows. Given M input channels, N output channels, and a convolution kernel, K, of size $d_k \times d_k$: For a pointwise convolution and a depthwise convolution, the number of parameters and the computational costs can be given as:

$$\begin{aligned} D_w &= d_k.d_k.(\alpha.M) \\ D_{wcnn} &= d_k.d_k.(\alpha.M).d_f.d_f \\ P_w &= (\alpha N).(\alpha M) \\ P_{wcnn} &= (\alpha N).(\alpha M).d_f.d_f, \end{aligned} \tag{5}$$

where D_w is the number of parameters of a depthwise convolution, the feature map's size is $d_f \times d_f$, the depthwise

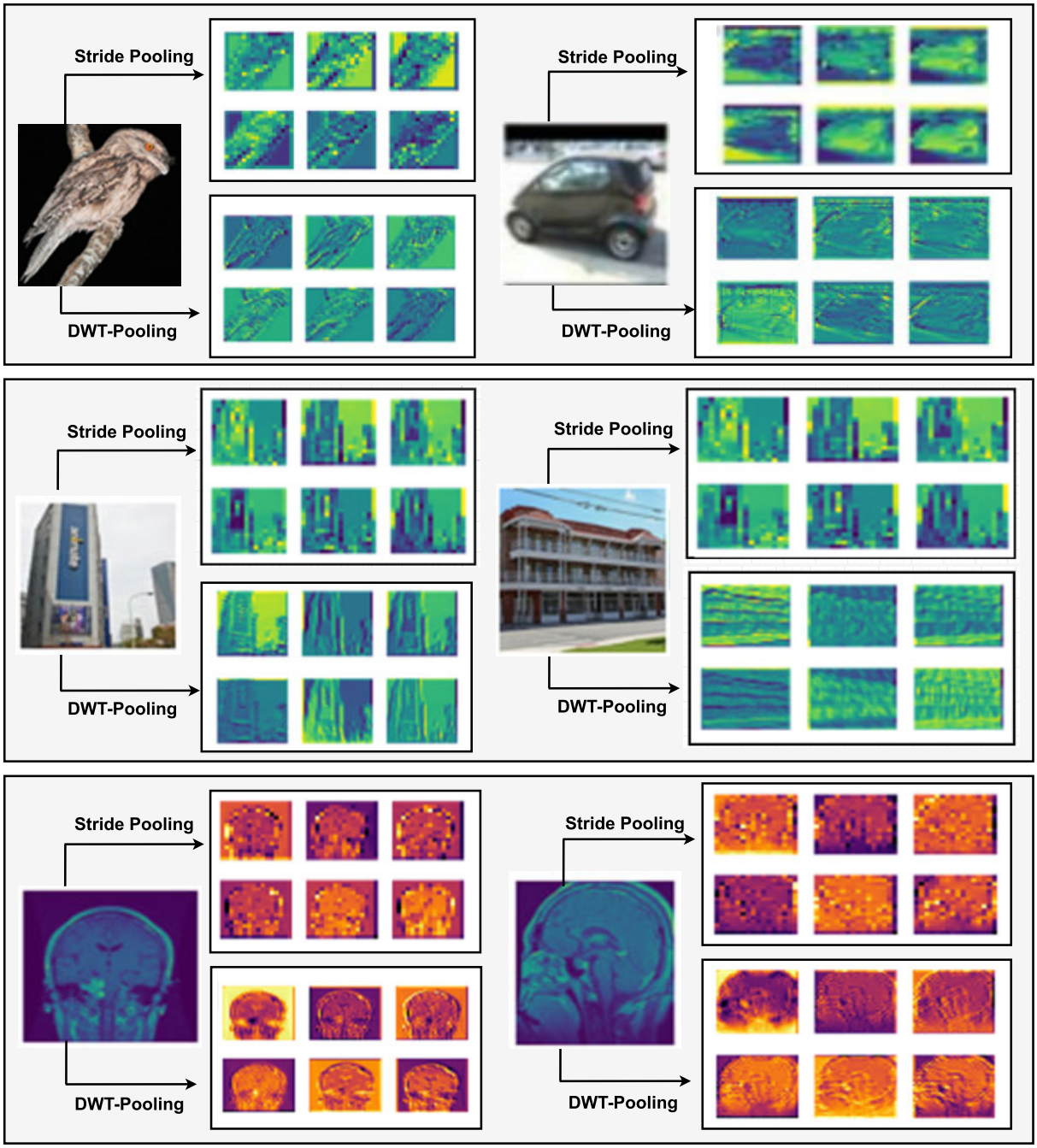

FIGURE 4. The feature maps of Standard MobileNets (top) and MobileNet-DWT (bottom) for medical and general objects using different datasets. In each subfigure, the first two rows shows the feature maps output from the original MobileNets, while the second two rows shows the output feature maps from wavelet pooling. Compared with base MobileNets, the feature maps of MobileNet-DWT are sharper and the object structures are more complete.

convolution's computational cost is D_{wcnn}, the number of parameters of a pointwise convolution is P_w, and the computational cost of a pointwise convolution is P_{wcnn}. The computational cost CC of a depthwise separable convolution with width multiplier α is:

$$CC = d_k.d_k.\alpha M.d_f.d_f + \alpha N.\alpha M.d_f.d_f, \tag{6}$$

where α is typically set to 1, 0.75, 0.5, and 0.25. In this work, we set the value of α by 1. It is worth mentioning that in our previous work [12], D^{DWT}_{WCNN} is represented as:

$$D^{DWT}_{WCNN} = DWT(D_{WCNN}), \tag{7}$$

where $DWT(\cdot)$ is the first level DWT of $\cdot$, and D_{WCNN} is the outcome of the depthwise operation. The D^{DWT}_{WCNN} is

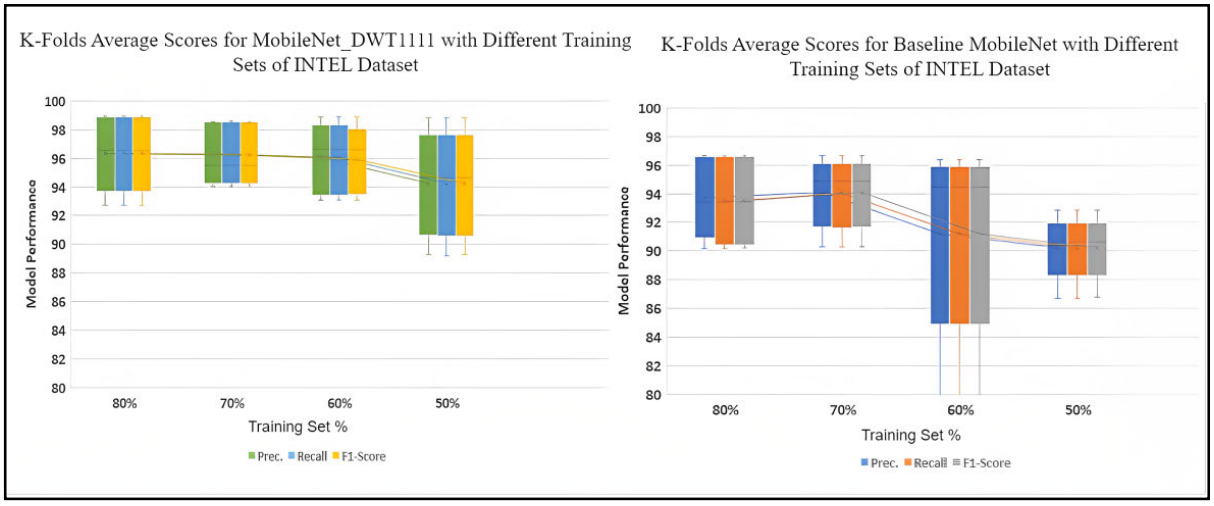

FIGURE 5. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using the Intel dataset.

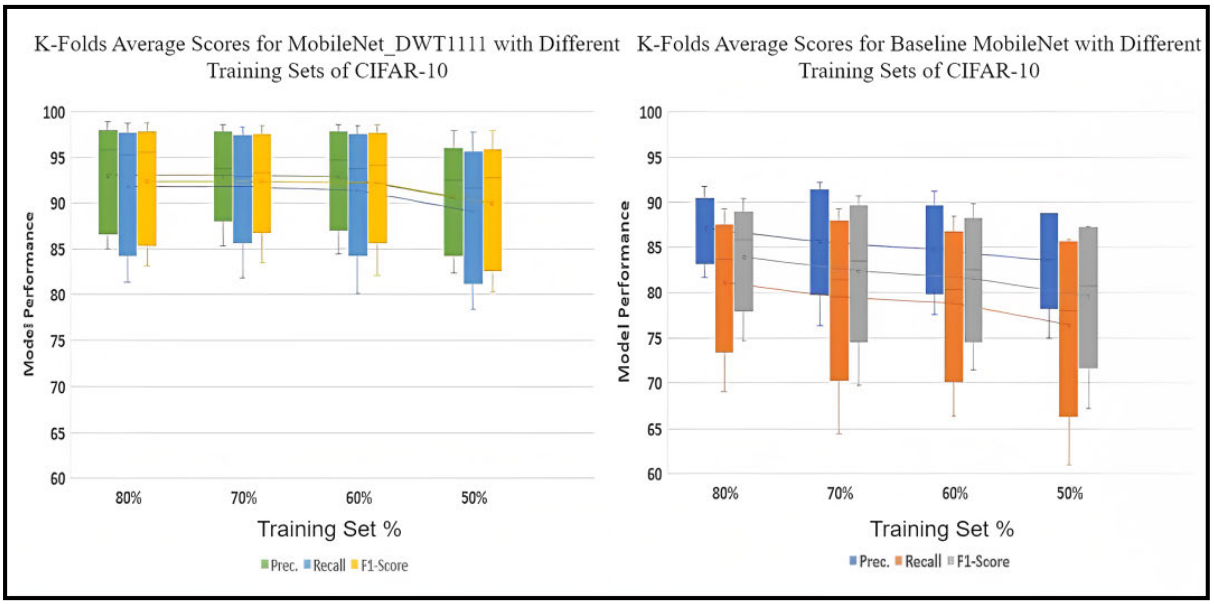

FIGURE 6. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using CIFAR-10 dataset.

defined as:

$$D_{WCNN}^{DWT} := \|(G_{LL}, G_{LH}, G_{HL}, G_{HH}) \tag{8}$$

where $\|$ denotes the concatenation operation, which means that the four sub-bands, i.e., LL, LH, HL, and HH, are included in the pooling stage. This is different from other wavelet pooling models in the literature that only considered the LL sub-band or other previous methods that considered certain sub-bands in an adaptive manner based on the input image [11].

B. SIMULATING VARYING DATA AVAILABILITY

As a standard supervised machine learning problem, given a training distribution of images X and a label distribution Y, our objective is to learn a classifier f_Θ, parameterized by a set of variables Θ, such that for any image $x \in X$ with corresponding label $y \in Y$, $y = f_\Theta(x)$. Let us suppose that our distribution is made of K different classes and that our training dataset, sampled from X and Y, is composed of N images per class. In the next sub-section, we show that the adopted datasets not only represent different areas of application but also contain different number of classes.

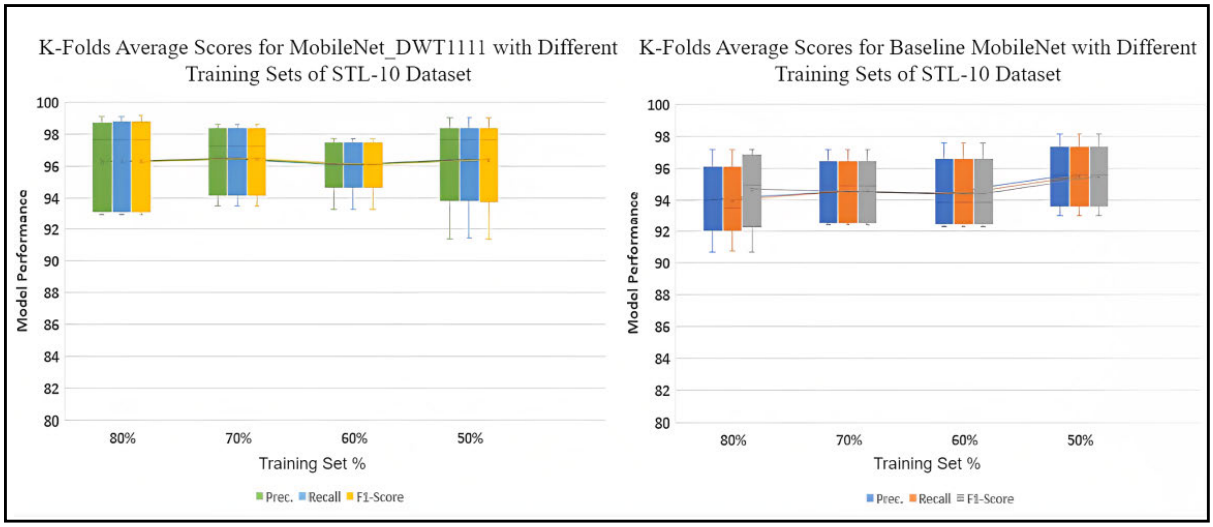

FIGURE 7. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using STL-10 dataset.

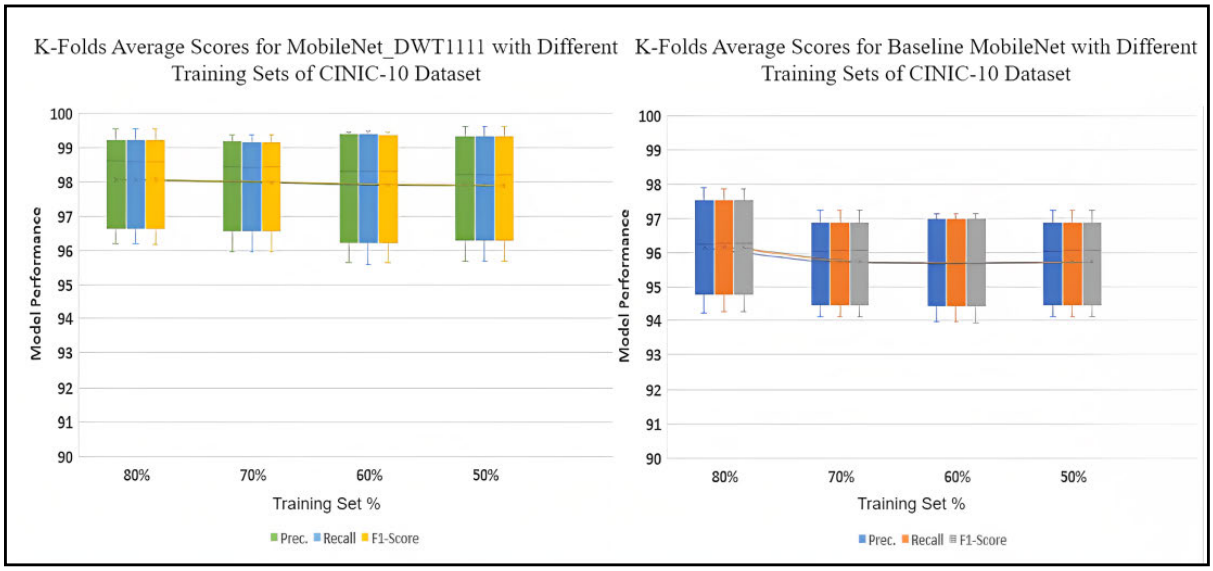

FIGURE 8. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using CINIC-10 dataset.

Our main research question is: *Given that model X adopts WP and model Y adopts traditional pooling, how much training data can be discarded, while training model X, until the model performance degrades (during testing) to that of model Y ?* To address this question, we have to train the model multiple times, each of which with a different amount of training data. Accordingly, we have to choose a range of variation for the amount of training data. The upper limit of this range is the whole amount of training data which is dictated by the initial 80-20 train-test split. This is depicted as *Training Set 1* in Fig. 2 which is 80% of the whole dataset. In the following lines, we explain how we decided the lower limit of the variation range.

For investigating the data efficiency of the model under consideration, we put three criteria for determining the variation range or thr *investigation range*, i.e., the range of training data sizes throughout which the model's performance is investigated. This includes the stopping condition, where we stop reducing further the size of the training data. These criteria are as follows:

1) The investigation range should include the case where the full training split is considered. This is why the ceiling of that range in our case is set to 80% of the training data–the whole training split.
2) The floor of the investigation range is determined by the point where the model starts to experience noticeable performance degradation.
3) Throughout the larger part of that range, the model's performance should not demonstrate significant variations. Criteria 2 and 3 are meant to ensure that data

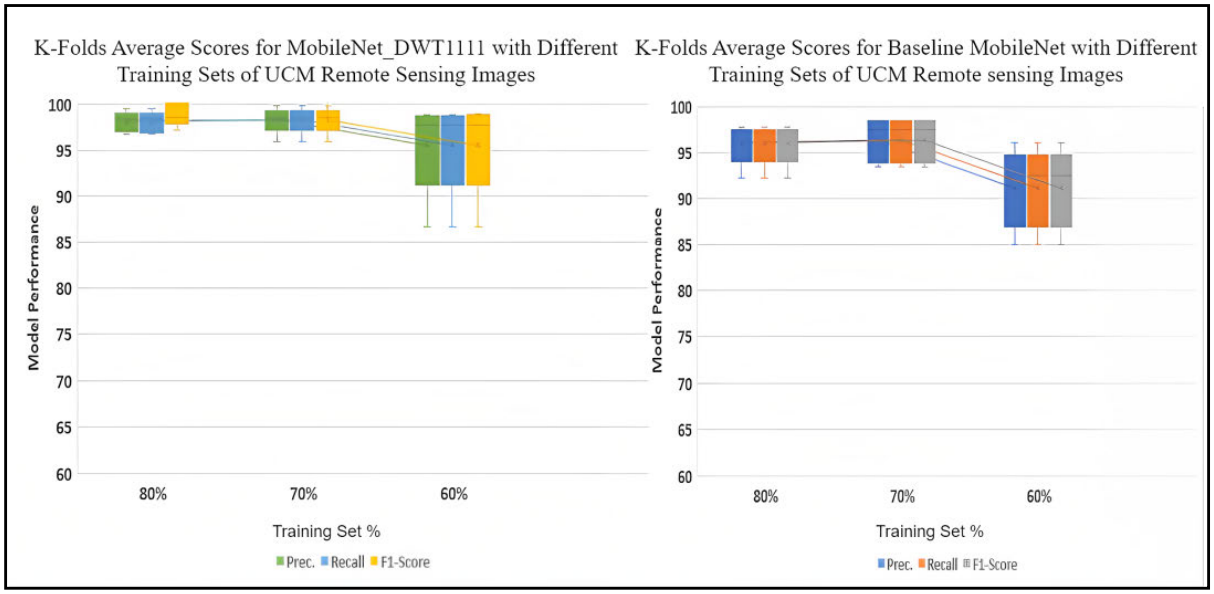

FIGURE 9. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using the UCM Land Scene dataset.

efficiency is not achieved at the expense of the model's performance, since this would constitute a degenerative type of training data saving.

We tested the model's performance on three datasets, namely, CIFAR-10, the Intel Image Dataset, and UCM Land Scene Remote Sensing dataset. According to the performance shown in Fig. 3 for the WP-base model, we decided to stop the investigation range at 50% of the whole dataset. From the whole training split (80% of the data), we constructed sub-samples using random sampling with sizes equal to 70%, 60%, and 50% of the whole dataset. This means that we chose four *checkpoints*, throughout the variation range, at which we investigated the training performance of a model with 10% spacing between each two checkpoints. This is depicted as *Training Set 2, 3, and 4* in Fig. 2.

C. DATASETS

We perform our experiments on seven widely used object recognition and diagnostic imaging datasets that are shown in Fig. 1, namely CIFAR-10 [25], CINIC-10 [26], STL-10 [27], INTEL [28], land scene remote sensing [29], Colon Disease [30], Malaria Cell [31], and Brain Tumor datasets [32], [33]. CIFAR-10, STL-10, and CINIC-10 are comprised of RGB images of dimension 32×32. The total number of categories is 10 and the classes represent different objects such as airplanes, cars, birds, cats, deers, dogs, frogs, horses, ships, and trucks. CIFAR-10 and CINIC-10 originally have 50,000 training images and 10,000 testing images with both sets balanced, while STL-10 has a $96 * 96$ image size. Also, it has 500 training images (10 pre-defined folds), and 800 test images per class.

Furthermore, the land scene remote sensing dataset (UCM) contains satellite images of 21 classes such as buildings, baseball fields, freeways, etc. The original size of the images is 256×256 pixels. Finally, for the medical classification datasets, we have the following: 1) the Brain Tumour dataset which has three classes of meningioma, glioma, and pituitary tumor brain type, 2) the WCE Colon images are captured via Wireless Capsule Endoscopy (WCE), which has four classes of normal, ulcerative colitis, polyps, and esophagitis type, 3) and the malaria cells dataset has two classes, infected and uninfected with a total of 27558 images. According to the aforementioned details, this research is validated on problems that are not just from different areas of appliation but also involve different number of classes. Figure 4 present a subjective comparison between the feature maps of standard MobileNets (top) and MobileNet-DWT (bottom) for medical and general objects from different datasets.

IV. RESULTS AND DISCUSSION

The results and simulations in this section were obtained using a machine with a GeForce RTX 2080 GPU (8 GB VRAM). All the computer programs were written in Python, and Tensorflow was incorporated as the backbone for the model training and testing. Table 1 summarizes the adopted performance metrics, where be referred, and TN refers to the number of True Positives, False Positives, False negatives, and True Negatives, respectively. We found them to be the most widely incorporated metrics in the relevant research on the classification of medical images. In the following lines, we motivate some key decisions in our simulations before presenting and discussing the results of our experiments. For the sake of results reproducibility, we provide the code of the testing stage publicly.[1]

In the rest of this section, we present a performance comparison between different flavors of the same deep

[1]https://github.com/shimaaelbana/Wavelet-Pooling-on-Improving-the-Data-Efficiency-of-Light-Weight-CNNs

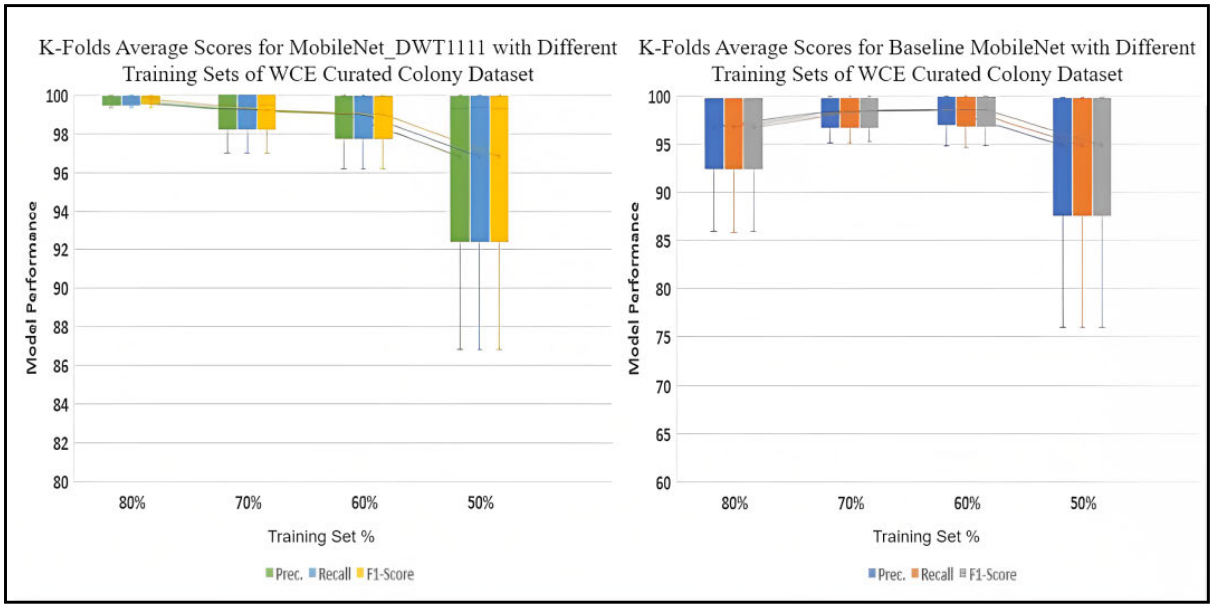

FIGURE 10. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using the WCE Curated Colon dataset.

TABLE 1. Quantitative Performance Measures.

Metric	Equation
Accuracy	(TP+TN)/(TP+FP+TN+FN)
Specificity	TN/(TN+FP)
Precision	TP/(TP+FP)
Recall=Senstivity	TP/(TP+FN)

model–MobileNet. Particularly, we compare the performance of the baseline MobileNet model with the WP-based MoileNet proposed in [12] which will referred to as *DWT1111* in the rest of this document. Whenever the term *classification performance* will be used, it will be referring to the classification accuracy. Lastly, five random samplings from the whole training dataset were acquired at each of the four checkpoints in the investigation range, and the results of each sampling iteration will be shown together with the average of all iterations. As mentioned in sec. I, this research addresses the evaluation of a model's data efficiency by addressing following research question: *Given that model X adopts WP and model Y adopts traditional pooling, how much training data can be discarded, while training model X, until the model performance degrades to that of model Y ?*

Computing the Achieved Percentage of Training Data Saving: In Table 2, we show a performance comparison between the baseline MobileNet and the WP-based MobileNet on the Intel Image Dataset at the four chosen checkpoints as mentioned earlier. When the training split is only 50% (instead of 80%), the WP-based model (DWT1111) achieved an average accuracy of 94.24%. On the other hand, When the training split is 80%, i.e., the full training split is used, the baseline model achieves an average accuracy

TABLE 2. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet-DWT1111 using various train-test split ratios on the Intel dataset.

Dataset	Intel Dataset			
%Data Split	80%		70%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	90.16	92.73	90.35	94.03
KF2	91.80	94.83	93.10	94.61
KF3	93.40	96.54	95.38	95.54
KF4	96.40	98.68	94.84	98.56
KF5	96.68	98.93	96.70	98.41
Average	93.069(±2.54)	**96.34(±2.34)**	94.07(±2.19)	**96.23(±1.90)**
%Data Split	60%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	90.80	93.02	86.76	89.20
KF2	79.08	94.03	90.00	92.13
KF3	95.25 a	96.59	90.64	94.68
KF4	96.38	97.69	90.90	96.38
KF5	94.49	98.94	92.87	98.82
Average	91.20(±6.34)	**96.06(±2.21)**	90.23(±1.98)	**94.24(±3.33)**

of 93.069%. This means that the WP-based model could outperform the baseline model while the training data size is 30% less with the former model. Hence, the training data saving in this case is more than 30%. This approach for computing the volume data saving will be adopted throughout the rest of this section. Although intuitive, we highlight the drawback of this approach in the next few lines.

Because we evaluate the performance of the models at discrete checkpoints only, which are 80%, 70%, 60%, and 50%, the *resolution* of measuring the volume of data saving is limited by those checkpoints. *Accordingly, the maximum saving that can be reported in this research is that the saving exceeds 30%, which is the difference between 80% and 50%.* Lastly, with every saving, we also report the accuracy gain (AG) which indicates how much training data can be saved

TABLE 3. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet-DWT1111 using various train-test split ratios on the CIFAR-10 dataset.

Dataset	CIFAR-10 Dataset			
%Data Split	80%		70%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	74.84	82.95	69.77	83.14
KF2	80.91	87.72	79.37	90.12
KF3	85.61	95.53	83.28	93.29
KF4	87.18	96.70	88.25	96.53
KF5	90.39	98.77	90.41	98.50
Average	83.79(±5.41)	**92.33(±6.0)**	82.22(±7.3)	**92.31(±5.4)**
%Data Split	60%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	71.56	81.82	67.59	79.91
KF2	77.58	88.99	76.07	98.98
KF3	82.37	94.11	80.57	92.05
KF4	86.26	96.72	86.94	93.72
KF5	89.86	98.50	87.02	97.85
Average	81.53(±6.4)	**92.03(±6.02)**	79.64(±7.3)	**89.70(±6.42)**

TABLE 4. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet- DWT1111 based Mobilenet using various train-test split ratios on STL-10 dataset.

Dataset	STL-10 Dataset			
%Data Split	80%		70%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	93.80	93.08	92.67	93.48
KF2	91.10	92.79	92.44	94.88
KF3	93.30	98.29	95.58	97.20
KF4	94.80	97.60	94.88	98.02
KF5	97.10	99.09	97.20	98.60
Average	94.02(±1.95)	**96.17(±2.6)**	94.55(±1.80)	**96.44(±1.94)**
%Data Split	60%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	93.85	93.28	94.33	91.50
KF2	92.28	96.14	96.50	96.16
KF3	92.71	96.15	93.00	97.66
KF4	95.57	97.14	95.49	97.67
KF5	97.57	97.71	98.16	99.00
Average	94.40(±1.95)	**96.08(±1.52)**	94.90(±1.183)	**96.40(±2.6)**

further before the gain turns into loss. In the same case in Table 2, the gain would be 1.17% (the difference between 94.24% and 93.069%). The higher the AG, the higher the data saving that can be achieved at the expense of the AG.

In Fig. 5, the whisker plot also compares the distribution of variations among the different folds for each of the adopted models. The WP-based model clearly shows less inter-quartile range (IQR) with regards to precision, recall, and F1-score at 80%, 70%, and 60%, in addition to achieving higher median values for the aforementioned performance metrics. It is worth mentioning that as the values resulting from the different folds approach a Gaussian distribution, the mean value approaches the median value. The only exception is at the 50% case where the WP-based model still achieves higher median yet the IQR among the different folds is higher. *The higher median values indicate that the training data saving which was reported in Table 2 (considering accuracy as the performance metric) generalizes well to other performance metrics*. Also, given that each whisker plot indicates the IQR of the distribution, comparing the *variance of IQRs*

TABLE 5. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet- DWT1111 based Mobilenet using various train-test split ratios on CINIC-10 dataset.

Dataset	CINIC-10 Dataset			
Data Split%	80%		70%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	94.23	96.20	94.47	95.97
KF2	95.34	97.12	95.69	97.21
KF3	96.27	98.57	96.16	98.42
KF4	97.13	98.83	96.97	98.91
KF5	97.86	99.55	97.46	99.38
Average	96.17(± 1.27)	**98.05(± 1.21)**	96.1(±1.03)	**98.0(±1.23)**
Data Split%	60%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	93.96	95.66	94.12	95.72
KF2	94.96	96.86	94.84	96.96
KF3	95.66	98.30	96.48	98.19
KF4	96.77	99.25	96.06	98.97
KF5	97.12	99.46	97.23	99.62
Average	95.69(±1.16)	**97.91(±1.44)**	95.74(±1.12)	**97.89(±1.40)**

TABLE 6. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet-DWT1111 using various train-test split ratios on land scene remote sensing dataset.

Dataset	Land-Use Scene (UCM) Dataset					
%Data Split	80%		70%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111	Baseline	DWT1111
KF1	92.23	96.80	94.13	95.96	89.06	86.83
KF2	97.80	97.19	97.48	98.58	92.48	95.83
KF3	95.76	99.57	98.48	98.69	84.90	98.88
KF4	97.33	98.52	98.37	99.84	96.08	97.70
KF5	97.23	98.57	93.50	98.53	93.29	98.69
Average	96.07(±2.03)	**98.13(±1.005)**	96.39(±2.14)	**98.32(±1.27)**	91.16(±3.84)	**95.59(±4.51)**

TABLE 7. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet-DWT1111 using various train-test split ratios on the WCE Curated Colon dataset.

Dataset	WCE Curated Colon Dataset			
%Data Split	80%		70%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	85.93	99.37	95.19	97.01
KF2	99.53	99.84	98.50	99.50
KF3	99.06	99.84	99.33	99.51
KF4	99.68	100.0	99.0	100.0
KF5	99.68	99.68	99.50	100.0
Average	96.78(±5.42)	**99.75(±0.21)**	98.0(±5.0)	**99.20(±1.11)**
%Data Split	60%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	94.69	99.31	99.65	98.45
KF2	99.65	99.82	75.51	99.82
KF3	99.14	96.06	99.65	99.82
KF4	99.31	100.0	99.82	100.0
KF5	100.0	99.82	99.82	96.81
Average	98.56(±1.95)	**99.00(±1.49)**	94.89(±9.62)	**96.98(±5.11)**

(for each performance metric separately) at varying sizes of training data shows that the WP-based model is less sensitive to variations in the volume of training data compared to the baseline model.

In Table 3, we show that on CIFAR-10, we can attain a data saving that exceeds 30% at an AG of 5.91%. Similarly, in Table 4 on the STL-10 dataset, it is shown that a data saving that surpasses 30% could be achieved at an AG of 2.38%. Table 5 on the CINIC-10 dataset features a data saving that is more than 30% at an AG of 1.72%. Figure 6 shows the distributions of the precision, recall, and F1-score, obtained from the five folds, which were attained on CIFAR-10 for the

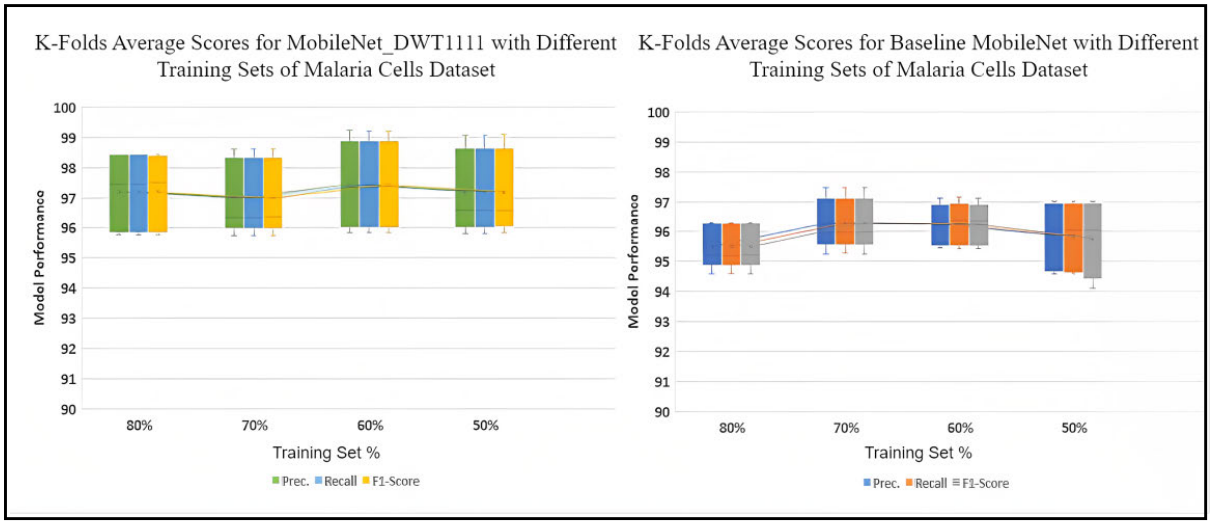

FIGURE 11. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using the Malaria Cell dataset.

TABLE 8. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet-DWT1111 using various train-test split ratios on the Malaria Cell dataset.

Dataset	Malaria Cell Dataset			
%Data Split	80%		70%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	95.22	95.77	95.92	95.72
KF2	94.64	95.95	95.25	96.39
KF3	96.28	97.42	96.11	96.35
KF4	96.30	98.42	96.71	98.03
KF5	95.31	98.36	97.45	98.59
Average	95.55(±0.64)	**97.18(±1.13)**	96.29(±0.74)	**97.02(±1.09)**
%Data Split	60%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111
KF1	95.67	96.27	94.57	96.28
KF2	95.43	95.84	94.77	95.80
KF3	96.35	97.34	96.04	96.55
KF4	96.64	98.48	96.83	98.13
KF5	97.14	99.22	97.01	99.07
Average	96.24(±0.626)	**97.43(±1.27)**	95.08(±1.01)	**97.17(±1.23)**

TABLE 9. A comparison between the classification accuracy attained by baseline MobileNet and MobileNet-DWT1111 based Mobilenet using various train-test split ratios on the brain tumor dataset.

Dataset	Brain Tumor Dataset					
%Data Split	80%		70%		50%	
KF/Models	Baseline	DWT1111	Baseline	DWT1111	Baseline	DWT1111
KF1	81.56	77.97	76.18	81.41	76.49	81.87
KF2	89.07	92.00	90.03	89.35	85.05	93.42
KF3	87.43	96.24	90.18	96.95	84.83	92.61
KF4	95.75	97.87	95.76	97.70	92.61	98.003
KF5	97.22	96.56	98.30	95.43	97.40	96.60
Average	90.21(±5.72)	**92.13(±7.34)**	90.09(±7.65)	**92.02(±5.99)**	87.48(±7.13)	**92.50(±5.67)**

models under consideration–the baseline MobileNet and the WP-based MobileNet. It can be seen that the WP-based model achieves higher median values for the precision, recall, and F1-score. Hence, the training data saving which was shown in Table 3 generalizes well to other performance metrics. Lastly, observing the *variance of IQRs* of the whisker plots in Fig. 6 at different percentages of training data highlights that the WP-based model is less sensitive to variations in the volume of training data compared to the baseline model. Similar insights can be drawn from Fig. 7 and Fig. 8. Particularly, higher median values of the 50% case for the WP-based than the 80% case for the baseline model indicates that a data saving volume similar to that achieved in Table 4 and Table 5 (considering accuracy as the performance metric) is reflected in other performance metrics. However, the sensitivity to data size variations is higher in the WP-based MobileNet than the baseline MobileNet for these two datasets–STL-10 and CINIC-10.

TABLE 10. A comparison between the Precision, recall, and F1-Score metrics attained by baseline MobileNet and DWT1111-based Mobilenet using various training ratios on different datasets.

Performance	Precision		Recall		F1-Score	
%Training/Model	DWT	Baseline	DWT	Baseline	DWT	Baseline
Dataset	CIFAR-10					
80%	93.03(±5.32)	87.10(±3.48)	91.84(±6.47)	81.10(±7.05)	92.41(±5.92)	83.90(±5.47)
70%	93.05(±4.72)	85.60(±5.65)	91.84(±5.83)	79.51(±8.81)	92.43(±5.30)	82.35(±7.43)
60%	92.93(±5.15)	84.75(±4.7)	91.47(±6.60)	78.80(±7.89)	92.17(±5.91)	81.57(±6.47)
Dataset	STL-10					
80%	96.26(±2.61)	94.04(±2.09)	96.6(±2.6)	94.04(±2.1)	96.2(±2.6)	94.05(±2.0)
70%	96.4(±1.95)	94.56(±1.78)	96.44(±1.94)	94.56(±1.80)	96.44(±1.90)	94.56(±1.78)
60%	96.06(±1.53)	94.40(±1.94)	96.07(±1.54)	94.4(±1.93)	96.06(±1.53)	94.5(±1.94)
Dataset	INTEL					
80%	96.35(±2.34)	93.70(±2.50)	96.36(±2.35)	93.70(±2.54)	96.30(±2.50)	93.72(±2.54)
70%	96.23(±1.91)	94.08(±2.19)	96.24(±1.90)	94.08(±2.20)	96.22(±1.90)	94.00(±2.1)
60%	96.05(±2.20)	91.21(±6.34)	96.05(±2.20)	91.22(±6.50)	96.10(±2.2)	91.21(±6.34)
Dataset	Brain Tumor					
80%	92.30(±7.05)	90.28(±6.0)	92.31(±7.0)	90.3(±6.0)	92.2(±7.01)	90.2(±6.0)
70%	91.99(±5.94)	89.78(±7.85)	92.0(±5.60)	89.79(±7.90)	91.9(±6.00)	87.79(±7.86)
60%	92.4(±5.6)	86.30(±7.10)	92.42(±5.62)	86.33(±7.13)	92.4(±5.6)	86.37(±7.10)
Dataset	Malaria Cells					
80%	97.20(±1.13)	95.51(±0.66)	97.21(±1.40)	95.51(±0.63)	97.20(±1.13)	95.50(±0.65)
70%	96.99(±1.10)	96.28(±0.75)	97.0(±1.12)	96.30(±0.76)	96.9(±1.10)	96.28(±0.74)
60%	97.43(±1.20)	96.30(±0.62)	97.44(±1.28)	96.24(±0.6)	97.4(±1.30)	96.24(±0.63)

Towards validating the previously mentioned results on a wide variety of datasets, we chose a remote sensing dataset in addition to three diagnostic imaging datasets. The latter family of datasets include a colon disease, malaria cell, and brain tumour datasets. Table 6 shows a performance comparison between the models under consideration on the Land Scene remote sensing dataset. Similar to all the previous datasets, we obtained a training data saving that exceeds 30%. The Colon Disease dataset (Table 7), the Malaria Cell datasets (Table 8), and the Brain Tumour dataset (Table 9) feature

TABLE 11. Comparing the performance of various classification techniques on CIFAR-10, STL-10, CINIC-10, INTEL, WCE Colon, Brain Tumor, Malaria Cells, and Land Scene (UCM) remote sensing images.

CIFAR-10		INTEL		STL-10		CINIC-10	
Model	**Acc.**	**Model**	**Acc.**	**Model**	**Acc.**	**Model**	Acc.
MobileNet-V2 [34]	79.10	Xception1 [35]	87.87	ResNet29_2x64d [26]	80.61	VGG-16 [26]	97.77
MobileNet-V1 [23]	79.64	Xception2 [35]	89.77	NAT-M4 [36]	92.61	ResNet-18 [26]	90.27
SqueezeNet [37]	82.36	Xception3 [35]	90.13	NAT-M2 [36]	97.2	DenseNet-121 [26]	91.26
Alexnet [38]	82.53	ResNet50_1 [35]	82.03	NAT-M3 [36]	97.80	ResNeXt29_2x64d [26]	91.45
EffNet [34]	83.20	ResNet50_2 [35]	87.13	MobileNet [23]	94.02	NAT-M1 [36]	93.4
DenseNet121 [37]	83.45	ResNet50_3 [35]	87.73	RegNet10B [39]	97.3	NAT-M2 [36]	94.1
MobileNet-DWT [11]	86.62	DenseNet169_1 [35]	84.57	VGG-19bn [40]	95.44	NAT-M3 [36]	94.3
ResNet-14 [41]	89.0	DenseNet169_3 [35]	87.83	FixMatch [42]	94.83	Efficient Ensembling [43]	95.064
Ours_DWT1111	**92.33**	Ours_DWT1111	**96.34**	Ours_DWT1111	**96.17**	Ours_DWT1111	**98.05**

WCE Colon		Brain Tumor		Malaria Cells		Land-Use Scene (UCM)		
Model	**Acc.**	**Model**	**Acc.**	**Model**	**Acc.**	**Model**	**Acc.(50%)**	**Acc.(80%)**
SVM [44]	94.83	$[AlexNet_{ft}]^0 + SVM$ [45]	88.35	VGG-16 [26]	95.85	PLSA(SIFT) [46]	76.55(±1.11)	71.38(±1.77)
MLP [44]	86.93	$[GoogleNet_{ft}]^0 + SVM$ [45]	88.69	VGG-19 [47]	95.92	GoogleNet [46]	92.70(±0.60)	94.31(±0.89)
InceptionResNetV2 [48]	80.13	$[ShuffleNet_{ft} + ShallowNet^0] + SVM$ [45]	91.62	Xception [47]	95.08	AlexNet [46]	93.98(±0.67)	95.02(±0.81)
ResNet50V2	84.25	$[ShuffleNet_{ft}]^{72} + SVM$ [45]	94.65	DenseNet-121 [47]	94.52	VGGNet-16 [46]	94.14(±0.69)	95.21(±1.20)
Xception	84.50	$[ResNet18_{ft}]^0 + SVM$ [45]	92.02	DenseNet-169 [47]	93.82	SPP with AlexNet [49]	94.77(±0.46)	96.67(±0.94)
DenseNet121	84.50	$[ResNet18_{ft} + ShallowNet]^0 + SVM$ [45]	92.14	DenseNet-201 [47]	90.54	TEX-Net with VGG [50]	94.22 ± (0.50)	95.31±(0.69)
InceptionV3	85.37	$[ResNet18_{ft}]^{56} + SVM$ [45]	96.76	Inception-V3 [47]	93.06	Gated Attention [51]	94.64 ± (0.43)	96.12 ± (0.42)
EfficientNetB0	88.25	$[ResNet18_{ft} + ShallowNet]^{56} + SVM$ [45]	97.25	ResNet-50 [47]	95.17	MIDC-Net CS [52]	95.41 ± (0.40)	97.40±(0.48)
MobileNetV2	92.25	CapsNet [53]	90.89	ResNet-101 [47]	95.62	RADC-Net [54]	94.79 ± (0.42)	97.05 ± (0.48)
MobileNet	96.78	Modified-CapsNet [55]	86.56	ResNet-152 [47]	95.05	VGG VD16 + SAFF [56]	–	97.02 ± (0.78)
MFuRe_CNN [48]	97.75	SVM [57]	91.14	SqueezeNet [47]	94.35	D-CNN with AlexNet [58]	–	96.67 ±(0.10)
WGS SVM (OACCorr) [44]	97.89	BMRI-Net-PFpM [59]	**98.45**	Proposed in [47]	96.82	MobileNet	91.16 ± (3.84)	96.07 ± (2.037)
Ours_DWT1111	**99.75**	Ours_DWT1111	92.5	Ours_DWT1111	**97.18**	Ours_DWT1111	**95.59 ± (4.51)**	**98.13 ± (1.005)**

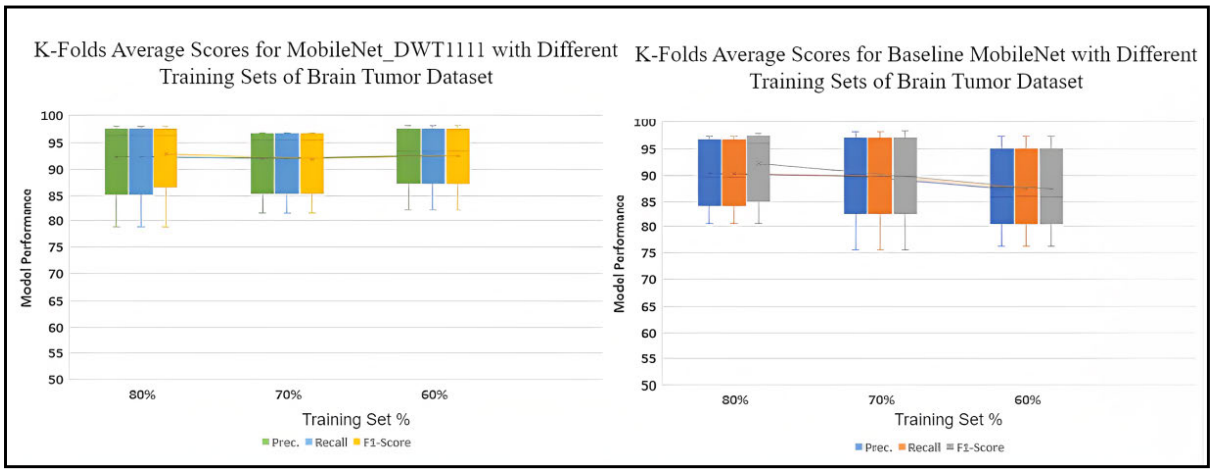

FIGURE 12. A comparison between the baseline and the DWT1111 MobileNet according to various performance metrics (precision, recall, and F1-score) using the brain tumor dataset.

similar results to the Land Scene dataset, in regard to data saving, with less than 2% accuracy gain in all the datasets. Finally, Table 10 presents a summary of the performance metrics on different datasets using various ratios of train-test splits.

We also considered the other performance metrics (precision, recall, and F1-score) on the remote sensing and medical datasets. In Fig. 9, the medians of the whisker plots at 60% for the WP-based model are less than those of the 80% case for the baseline model, so the data saving advantage of the WP-based model does not apply to other performance metrics for this dataset. By observing the median scores at 70% for the DWT1111 model, we report an achievable data saving that is more than 10% yet less than 20%. Also, the sensitivity to dataset size variations is higher for the WP-based model than the baseline model.

Figure 10 shows the results of the colon dataset. The saving lies between 20% and 30% for the three depicted performance measures while the sensitivity to dataset size variation of the WP-base model is higher than that of the baseline model. On the malaria dataset of Fig. 11, the data saving of the WP-based model is at an advantage of more than 30% compared to the baseline model. The results on the brain tumour dataset (Fig. 12) shows a data saving that is approximately equal to 20%. The sensitivities to dataset size variation for the two models, on the malaria and brain tumour datasets, are comparable.

V. CONCLUSION

In this research, we highlighted the potential of wavelet pooling with regard to improving the training data efficiency of MobileNets. Wavelet pooling was shown to improve the performance of deep network models in classification and segmentation applications. However, its impact on reducing the amount of training data required to attain a certain performance level, e.g., classification accuracy level, had not

been explored before. To the best of our knowledge, this is the first research to investigate the volume of data that can be saved while training a WP-based model before the model's performance levels or degrades to the performance of another model that adopts conventional pooling. Since the performance of the model might vary considerably with the distribution of the dataset under consideration, we ran our simulations on seven widely adopted datasets for diagnostic imaging and object recognition. Using a flavor of wavelet pooling from the recent literature, we have shown an average training data saving that exceeds 30% when the classification accuracy is adopted as the metric of performance. When other metrics such as precision, recall, and F1-score are adopted, object recognition datasets feature data savings that are similar to the case where accuracy is considered, i.e., more than 30%. Less data savings though, that exceeds 22%, were achieved for diagnostic imaging datasets. Our choice to focus on a light-weight architecture, that stands out where testing resources are scarce, aimed to stress the potential of WP in applications with scarce training resources as well. The findings of this research have significance to other areas of research that involve constrained learning resources including edge computing and green AI. Future research directions may investigate the generalizability of our results to other datasets, in addition to comparing the attained data efficiency to that of larger deep models.

ACKNOWLEDGMENT

(Shimaa El-Bana, Ahmad Al-Kabbany, and Said El-Khamy contributed equally to this work.)

REFERENCES

[1] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.
[2] S. Kum, S. Oh, J. Yeom, and J. Moon, "Optimization of edge resources for deep learning application with batch and model management," *Sensors*, vol. 22, no. 17, p. 6717, Sep. 2022.
[3] J. Xu, W. Zhou, Z. Fu, H. Zhou, and L. Li, "A survey on green deep learning," *CoRR*, vol. abs/2111.05193, pp. 1–61, Nov. 2021.
[4] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J. Zhu, "Dataset distillation by matching training trajectories," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10708–10717.
[5] G. N. Karystinos and D. A. Pados, "On overfitting, generalization, and randomly expanded training sets," *IEEE Trans. Neural Netw.*, vol. 11, no. 5, pp. 1050–1057, 2000.
[6] S. Yang, Z. Xie, H. Peng, M. Xu, M. Sun, and P. Li, "Dataset pruning: Reducing training data by examining generalization influence," 2022, *arXiv:2205.09329*.
[7] B. Ghojogh and M. Crowley, "Principal sample analysis for data reduction," in *Proc. IEEE Int. Conf. Big Knowl. (ICBK)*, Nov. 2018, pp. 350–357.
[8] J. M. Fortuna-Cervantes, M. T. Ramírez-Torres, M. Mejía-Carlos, J. S. Murguía, J. Martinez-Carranza, C. Soubervielle-Montalvo, and C. A. Guerra-García, "Texture and materials image classification based on wavelet pooling layer in CNN," *Appl. Sci.*, vol. 12, no. 7, p. 3592, 2022.
[9] A. de Souza Brito, M. B. Vieira, M. L. S. C. de Andrade, R. Q. Feitosa, and G. A. Giraldi, "Combining max-pooling and wavelet pooling strategies for semantic image segmentation," *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115403.
[10] Q. Li and L. Shen, "3D WaveUNet: 3D wavelet integrated encoder–decoder network for neuron segmentation," 2021, *arXiv:2106.00259*.
[11] S. El-Khamy, A. Al-Kabbany, and S. El-Bana, "Less is more: Matched wavelet pooling-based light-weight CNNs with application to image classification," *IEEE Access*, vol. 10, pp. 59592–59602, 2022.
[12] S. El-Khamy, A. Al-Kabbany, and S. El-Bana, "Going shallower with MobileNets: On the impact of wavelet pooling," in *Proc. 38th Nat. Radio Sci. Conf. (NRSC)*, vol. 1, Jul. 2021, pp. 126–138.
[13] S. E. El-Khamy, A. Al-Kabbany, and S. El-Bana, "MLRS-CNN-DWTPL: A new enhanced multi-label remote sensing scene classification using deep neural networks with wavelet pooling layers," in *Proc. Int. Telecommun. Conf. (ITC-Egypt)*, Jul. 2021, pp. 1–5.
[14] P. Liu, H. Zhang, W. Lian, and W. Zuo, "Multi-level wavelet convolutional neural networks," *IEEE Access*, vol. 7, pp. 74973–74985, 2019.
[15] T. Williams and R. Li, "Wavelet pooling for convolutional neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.
[16] R. Mall, V. Jumutc, R. Langone, and J. A. K. Suykens, "Representative subsets for big data learning using k-NN graphs," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2014, pp. 37–42.
[17] O. Bachem, M. Lucic, and A. Krause, "Practical coreset constructions for machine learning," 2017, *arXiv:1703.06476*.
[18] A. Lapedriza, H. Pirsiavash, Z. Bylinskii, and A. Torralba, "Are all training examples equally valuable?" 2013, *arXiv:1311.6510*.
[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.
[20] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.
[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[22] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019.
[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 7, 2009.
[26] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "CINIC-10 is not ImageNet or CIFAR-10," 2018, *arXiv:1810.03505*.
[27] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
[28] *Intel Image Classification Dataset*. Accessed: May 30, 2023. [Online]. Available: https://www.kaggle.com/datasets/puneet6060/intel-image-classification
[29] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2010, pp. 270–279.
[30] *WCE Curated Colon Disease Dataset*. Accessed: May 30, 2023. [Online]. Available: https://www.kaggle.com/datasets/francismon/curated-colon-dataset-for-deep-learning
[31] *Malaria Cell Images Dataset*. Accessed: May 30, 2023. [Online]. Available: https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria
[32] *Brain Tumor MRI Dataset*. Accessed: May 30, 2023. [Online]. Available: https://www.kaggle.com/datasets/denizkavi1/brain-tumor?select=1
[33] J. Cheng et al., "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *PLoS One*, vol. 10, no. 10, Oct. 2015.
[34] I. Freeman, L. Roese-Koerner, and A. Kummert, "EffNet: An efficient structure for convolutional neural networks," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 6–10.
[35] M. Rahimzadeh, S. Parvin, E. Safi, and M. R. Mohammadi, "Wise-SrNet: A novel architecture for enhancing image classification by learning spatial resolution of feature maps," 2021, *arXiv:2104.12294*.
[36] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, and V. N. Boddeti, "Neural architecture transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 2971–2989, Sep. 2021.
[37] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, and M. A. Ayidzoe, "RMAF: ReLU-memristor-like activation function for deep learning," *IEEE Access*, vol. 8, pp. 72727–72741, 2020.
[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[39] P. Goyal, Q. Duval, I. Seessel, M. Caron, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski, "Vision models are more robust and fair when pretrained on uncurated images without supervision," 2022, *arXiv:2202.08360*.
[40] H. M. D. Kabir, M. Abdar, A. Khosravi, S. M. J. Jalali, A. F. Atiya, S. Nahavandi, and D. Srinivasan, "SpinalNet: Deep neural network with gradual input," *IEEE Trans. Artif. Intell.*, early access, Jun. 21, 2022, doi: 10.1109/TAI.2022.3185179.
[41] E. Pishchik, "Trainable activations for image classification," Preprints.org, 2023, Art. no. 2023010463, doi: 10.20944/preprints202301.0463.v1.
[42] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 596–608.
[43] A. Bruno, D. Moroni, and M. Martinelli, "Efficient adaptive ensembling for image classification," 2022, *arXiv:2206.07394*.
[44] S. Suman, F. Hussin, A. Malik, S. Ho, I. Hilmi, A. Leow, and K.-L. Goh, "Feature selection and classification of ulcerated lesions using statistical analysis for WCE images," *Appl. Sci.*, vol. 7, no. 10, p. 1097, Oct. 2017.
[45] C. Öksüz, O. Urhan, and M. K. Güllü, "Brain tumor classification using the fused features extracted from expanded tumor region," *Biomed. Signal Process. Control*, vol. 72, Feb. 2022, Art. no. 103356.
[46] G. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.
[47] A. Maqsood, M. S. Farid, M. H. Khan, and M. Grzegorzek, "Deep malaria parasite detection in thin blood smear microscopic images," *Appl. Sci.*, vol. 11, no. 5, p. 2284, Mar. 2021.
[48] F. J. P. Montalbo, "Diagnosing gastrointestinal diseases from endoscopy images through a multi-fused CNN with auxiliary layers, alpha dropouts, and a fusion residual block," *Biomed. Signal Process. Control*, vol. 76, Jul. 2022, Art. no. 103683.
[49] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sens.*, vol. 9, no. 8, p. 848, Aug. 2017.
[50] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, "Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 74–85, Apr. 2018.
[51] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2127–2136.
[52] Q. Bi, K. Qin, Z. Li, H. Zhang, K. Xu, and G. Xia, "A multiple-instance densely-connected ConvNet for aerial scene classification," *IEEE Trans. Image Process.*, vol. 29, pp. 4911–4926, 2020.
[53] P. Afshar, K. N. Plataniotis, and A. Mohammadi, "Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1368–1372.
[54] Q. Bi, K. Qin, H. Zhang, Z. Li, and K. Xu, "RADC-Net: A residual attention based convolution network for aerial scene classification," *Neurocomputing*, vol. 377, pp. 345–359, Feb. 2020.
[55] P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain tumor type classification via capsule networks," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 3129–3133.
[56] R. Cao, L. Fang, T. Lu, and N. He, "Self-attention-based deep feature fusion for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 43–47, Jan. 2021.
[57] J. Cheng, W. Huang, S. Cao, R. Yang, W. Yang, Z. Yun, Z. Wang, and Q. Feng, "Enhanced performance of brain tumor classification via tumor region augmentation and partition," *PLoS ONE*, vol. 10, no. 10, Oct. 2015, Art. no. e0140381.
[58] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.
[59] A. Mondal and V. K. Shrivastava, "A novel Parametric Flatten-p Mish activation function based deep CNN model for brain tumor classification," *Comput. Biol. Med.*, vol. 150, Nov. 2022, Art. no. 106183.

SHIMAA EL-BANA (Student Member, IEEE) received the B.S. degree in electronics and communication engineering from the Alexandria Higher Institute of Engineering and Technology (AIET), Alexandria, Egypt, in 2013, and the M.S. degree in electrical engineering from the Arab Academy for Science and Technology, Alexandria, in 2020. She is currently pursuing the Ph.D. degree in electrical engineering at Alexandria University, Egypt. She is also a Teaching Assistant with the Department of Electronics and Communication Department, AIET. Her research interests include image processing, imaging diagnostics, bio-signal analysis, and machine learning.

AHMAD AL-KABBANY (Member, IEEE) received the bachelor's degree in electronics and communications engineering from the Arab Academy for Science and Technology (AAST), Egypt, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Ottawa, Canada, in 2016.

From 2006 to 2010, he was a Research and Teaching Assistant with the Electronics and Communications Engineering Department, AAST. In 2010, he was a Graduate Research Intern with the University of Ulster, Northern Ireland. From 2010 to 2016, he was a Graduate Research Assistant and a Postdoctoral Research Fellow with the School of Electrical Engineering and Computer Science, University of Ottawa. During the Ph.D. studies, he co-developed an image matting techniques that was placed eighth worldwide as per a widely adopted benchmark in the field. In 2016, he was an Assistant Professor and then, he has been an Associate Professor in electronics and communications engineering with AAST, since 2020, where he also co-founded the Intelligent Systems Laboratory. His research interests include the applications of machine learning and graph optimization in N-D signal processing, analysis, synthesis, and communication, in addition to communication networks. He also co-founded VRapeutic–a UNICEF Innovation Fund portfolio software house that develops therapeutic, AI-enabled solutions using virtual reality technology, with a focus on developmental challenges.

HASSAN M. ELRAGAL (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Alexandria, Egypt, in 1991 and 1995, respectively, and the Ph.D. degree in electrical engineering from Southern Methodist University, Dallas, TX, USA, in July 2001. His research interests include signal processing, fuzzy logic, neural networks, genetic algorithms, and particle swarm optimization.

SAID EL-KHAMY (Life Fellow, IEEE) received the Ph.D. degree from the University of Massachusetts, Amherst, USA, in 1971. Currently, he is an Emeritus Professor with the Electrical Engineering Department, Faculty of Engineering, Alexandria University, Egypt. His research interests include modern signal processing techniques, wireless communications, smart antenna arrays, cognitive radio, UWB, and image processing. He has published more than 300 scientific publications and he also chaired technical sessions in several international conferences. He is a fellow of the Electromagnetic Academy. He has also earned several international awards among which are the IEEE's Antennas and Propagation Society R. W. P. King Best Paper Award, in 1980, and the IEEE Region 8 Volunteer Award, in 2011.