A Frequency-Domain Approach with Learnable Filters for Image Classification

Article *in* SSRN Electronic Journal · January 2023

DOI: 10.2139/ssrn.4333420

4 authors:

Jose Augusto Stuchi
University of São Paulo
17 PUBLICATIONS **163** CITATIONS
SEE PROFILE

Natalia Canto
University of Campinas
6 PUBLICATIONS **1** CITATION
SEE PROFILE

Levy Boccato
University of Campinas
45 PUBLICATIONS **389** CITATIONS
SEE PROFILE

Romis Attux
University of Campinas
330 PUBLICATIONS **1,665** CITATIONS
SEE PROFILE

A frequency-domain approach with learnable filters for image classification

José Augusto Stuchi[a], Natalia Gil Canto[b,*], Levy Boccato[b], Romis Attux[b]

[a]*Phelcom Technologies, Rua José Missali, 820, São Carlos, 13562-405, SP, Brazil*
[b]*School of Electrical and Computer Engineering, University of Campinas, Av. Albert Einstein, 400, Campinas, 13083-852, SP, Brazil*

Abstract

Machine learning applied to computer vision and signal processing is achieving results comparable to the human brain due to the great improvements brought by deep neural networks (DNN). The majority of state-of-the-art architectures are DNN related, but only a few explicitly explore the frequency domain to extract useful information and improve the results. This paper presents a new approach for exploring the Fourier transform of the input images, which is composed of trainable frequency filters that boost discriminative components in the spectrum. Additionally, we propose a cropping procedure to allow the network to learn both global and local spectral features of the image blocks. The proposed method proved to be competitive with respect to well-known DNN architectures in the selected experiments, which involved texture classification, cataract detection and retina image analysis, where there is a noticeable appeal for the frequency domain, with the advantage of being a lightweight model.

Keywords: machine learning, Fourier analysis, image classification, deep learning, frequency filtering

1. Introduction

The last decade witnessed extraordinary developments in the field of machine learning [1]. The popularization of deep learning (DL) methods, in

*Corresponding author
Email address: nataliacantto@gmail.com (Natalia Gil Canto)

particular, led to performance improvements in some of the most emblematic problems in artificial intelligence, like pattern recognition [2], computer vision [3], audio processing [4], natural language processing [5] and, also, problems involving real-time game strategies [6].

In these application domains, the effective use of machine learning techniques classically depended on the existence of a suitable pre-processing stage, in which the raw data could be transformed into more "explicit" features, typically with a reduced dimensionality when compared to the original data. This approach requires a careful design of the feature extractor by experts with domain knowledge, in order that meaningful features be passed on to the classifier/predictor.

On the other hand, the approach explored in deep learning is to let the model learn directly from the raw data. In this sense, the deep model creates efficient internal representations for the data in the process of learning how to perform the desired task. Additionally, the "deeply layered" structure of the model has the potential of engendering a hierarchical representation, the first layers extracting low level features that are progressively combined into high level features by the subsequent layers. Interestingly, this possibility becomes actual if sufficiently large datasets and enough computational power are available [7, 8].

Deep learning ignited a revolution when systematically employed in the design of convolutional neural networks (ConvNets). These networks, proposed in the 1980s by LeCun [9], are inspired by properties of the human visual cortex [1]. The key feature of a ConvNet is the use of convolutional kernels that, together with a nonlinear activation function, process the input information. Additional stages, such as pooling, also play an important role.

There has been an extensive investigation on theoretical and practical aspects of convolutional neural networks [8]. From an implementation perspective, it is clear that computational performance depends heavily on the availability of an efficient algorithm for performing convolution. In the context of linear machine learning, an important possibility was raised decades ago in the context of adaptive filtering: to perform filter adaptation in the frequency domain. As discussed in [10], expressive computational gains were reached as the number of filter coefficients was increased. With the steady improvement of hardware, this idea has never become dominant in the field, but, interestingly, similar notions took form in the more complex case of ConvNets, as discussed in [11].

In this context, other important questions naturally arise, such as: what

would be the model of a frequency-domain layer? How can we efficiently train a layer of this kind? Is it possible to extract both global and local frequency-domain information from the input data (images)? And, finally, what are the advantages of explicitly exploring the frequency domain, instead of using the well-established ConvNets?

In addition to the aforementioned computational aspects, another important motivation for such a model comes from the idea that, depending on the problem, useful properties of the input data can be more easily perceived and extracted in the frequency domain. For instance, when classifying different plantation types on images captured by a drone, probably the frequency representation will bring more useful information due to the periodic nature of the data. In these images, high frequency components represent fine details, such as plant borders and corners, whereas low frequency components encode large scale information, such as shape and homogeneous areas of the plants and soil. Additionally, intermediate concentrations of frequencies in certain 2D spectrum regions will represent the periodic spacing between plants in such crops. Although the use of frequency-based features may appear to be in contradiction with the design philosophy of deep learning, this is not case, since the availability of large high-quality datasets is not guaranteed in many applications of interest.

The present work addresses all these questions and presents a new neural network approach that explores frequency-domain layers and can be applied to different computer vision problems. The proposed model is characterized by layers of frequency filters, which extract features from the magnitude spectrum of the image, and whose parameters are adjusted with the aid of gradient-based optimization methods using the error backpropagation algorithm.

The contribution presented here can be considered an extension of preliminary ideas brought in [12] and, independently, in [9]. In [12], fixed frequency-domain filters were used along with a cropping procedure, which splits the image in blocks, in order to capture global and local frequency components. Here, the frequency filters for each block are fully trained using the back-propagation algorithm, allowing the proposed network to learn the most discriminative frequency components of the data and to achieve excellent results, comparable to those obtained by renowned ConvNets. Moreover, the number of adjustable weights of the network was significantly reduced by adopting a radial-based filtering approach, in which frequencies located at the same radial distance from the spectrum center share the same weight.

It is important to remark that Fourier analysis is widely applied together with ANNs for decades and in different applications, such as brain electroencephalogram processing [13], cardiovascular analysis [14], speech processing [15], seismic analysis [16] and face spoofing detection [12]. Besides, it is also explored with deep learning, as, for example, in the speech processing field [17, 18], where an audio signal is converted into a 2D spectrogram image.

Most of these works transform the input data to a different domain (e.g., frequency) and, then, feed the classification model with the extracted features, thus exploring a more traditional approach in machine learning. Other recent works explore frequency domain operations in the context of representation learning, as will be described in the next section, but they usually apply the inverse transform before generating the final decision.

In this context, our proposal follows a different path since it does not use handcrafted features extracted from the frequency domain, but it allows the network to learn frequency filters that boost the most discriminative components for that specific classification problem. Moreover, our method does not perform successive direct and inverse Fourier transformations, but works solely in the frequency domain by processing the Fourier magnitude spectrum, thus simplifying the training and inference operations.

This paper is organized as follows: Section 2 presents a summary of the related literature, mainly the works that combine representation learning and frequency domain. Section 3 describes the proposed frequency-domain approach in detail, as well as the method used to train the network in the frequency domain. Section 4 brings the computational experiments performed in different datasets, like texture and retina images, with the purpose of evaluating the proposed method. In this section, the results and comparative analysis with deep learning architectures are also presented in order to emphasize the real contribution of the proposed method. Finally, Section 5 presents the conclusions and perspectives for future research and applications.

2. Related Work

Using the frequency domain in the context of representation learning and deep networks is relatively recent, most works having been published in the last ten years. Without the intention of covering all the literature in this

area, the following works are a list of important efforts related to the scope of this paper.

In [11], the authors brought an efficient way to train ConvNets in the frequency domain, reducing the training time. Indeed, the authors showed that when the ConvNet has many feature maps, processing in the frequency allows to accelerate the training and inference processes. In this approach, the Fourier transform of the image and the convolutional kernel are computed and, then, multiplied in the frequency domain. Then, the inverse Fourier transform is applied, returning the feature map to the spatial domain for further processing by the next layer. In this way, part of the network is in the frequency and the inverse 2D discrete Fourier transform (2D-DFT) is employed, which imposes an increase in computational cost, especially when the architecture has many layers.

In [19], the authors presented a way to train deep belief convolutional networks in frequency with the purpose of reducing the time required to compute the convolutions. Using the Fourier transform to optimize convolution operations, the authors observed a speed gain of 8 times in 2D images (ImageNet [20]) and 200 times in 3D volumes (OASIS dataset [21]). Basically, the convolution operations are performed in the frequency domain and, then, the inverse Fourier transform is applied before passing the signals to the subsequent layers, which, therefore, process feature maps in the spatial domain.

The work of [22] presented a strategy for training ConvNets filters directly in the frequency domain using the backpropagation algorithm. In addition, the authors presented a spectral clustering technique to reduce the dimensionality of the data. Interestingly, it was observed that this technique can help to preserve more relevant information when compared to clustering approaches commonly employed in ConvNets, such as max-pooling. The proposed method was applied to classification problems and showed competitive results for datasets such as CIFAR-10/100 [23]. However, the network processing is not entirely conceived in the frequency domain: although the 2D-DFT is computed at the input and the filters are adjusted in the frequency domain, the inverse 2D-DFT is applied after filtering, bringing the information back to the spatial domain, which makes the method more computationally costly.

In [24], the authors proposed a way of compressing ConvNets using discrete cosine transform (DCT), focusing on a smaller amount of weights and their underlying connections. In this method, a large number of low energy

DCT coefficients can be discarded, increasing the network compression without compromising its accuracy. The main focus of the work is on reducing the number of parameters of ConvNets in order to allow execution in mobile environments. However, in the same way as the previous works, it still applies a mixed network switching between the space and frequency domains, performing inverse 2D-DCT transformations.

In [25], the authors proposed a method for training a convolutional network completely in the frequency domain, also exploring the Fourier transform. In particular, they described a way to train networks without performing inverse transforms in the intermediate layers, which reduced the training time and did not compromised the performance. Experiments carried out on MNIST [26] and CIFAR-10 datasets showed that the approach achieved performances close to those of usual ConvNets. Nevertheless, when connecting the frequency layer to fully connected layers, the inverse Fourier transform is applied so that the feature maps return to the spatial domain for posterior processing by the last layers of the network.

Another method for compressing ConvNets was proposed in [27], but now through dynamic pruning of non-important coefficients of the frequency domain, obtained through the 2D-DCT. Basically, the convolution operations are mapped to the frequency domain, where dynamically pruning of spectral coefficients also takes place. Then, the inverse 2D-DCT transform is calculated so that the feature map returns to the spatial domain, serving as an input for the next layer. The compression obtained with this method, however, is very interesting, greatly reducing the number of ConvNet parameters.

In [28], the authors presented a convolutional network based on spectral operations, bringing a new activation function called sReLU (Spectral Rectified Linear Unit), which allows computations to be performed in the frequency domain (Fourier), avoiding multiple transformations between space and frequency domains. In addition, they explored only lower frequency components, making a fusion between convolutional layers and subsampling layers, also presenting a technique of batch normalization in frequency. The results for classification of facial images from the AT&T database [29] and numbers recognition of MNIST dataset, using the LeNet-5 [30] network, adapted for frequency, were very interesting, showing gains in accuracy, as well as faster training and network operation when compared with the spatial approach. Although this work aimed at implementing most of the network processing in the frequency domain, the inverse Fourier transform is still computed before entering the fully connected layers.

A new framework for training ConvNets in the frequency domain was presented in [31], exploring the ideas of linear exponential unit and pyramidal pooling, both in the frequency domain. Despite the fact that the training of the convolutional layers occurs entirely in the frequency domain, the proposed method applies the Fourier inverse transform when leaving these layers and also during the backpropagation phase.

More recently, the paper [32] presented a strategy of pre-processing images in the frequency, which aims to identify the trivial spectral components that can be removed without loss of accuracy. This strategy is applied before the images are provided as input to a standard ConvNet. The experiments showed that frequency learning can achieve better accuracy than conventional spatial architectures, since it performs subsampling in the data, reducing the size of the image to be used as input in the network. The proposed method showed gain in relation to spatial architectures such as ResNet-50[33], MobileNet-v2[34] and *Mask* R-CNN [35] on datasets such as ImageNet and COCO [36]. Instead of resizing the image in the spatial domain, the authors apply 2D-DCT to images in the YCbCr format and use the transformed coefficients as input for training and inference operations in the ConvNet. In addition, they presented a way to select channels in the frequency in order to remove trivial spectral components, improving and speeding up the inference. Therefore, the work combines pre-processing in the frequency domain with state-of-the-art spatial processing via convolutional networks.

In light of the presented literature review, it is possible to verify that these papers apply a frequency transformation on the data (Fourier or cosine) and, in the sequence, process this information for some machine learning application. In some methods, the frequency domain is combined with state-of-the-art ConvNets. Additionally, the inverse transform is applied in almost all approaches, in order to bring the information back to the spatial domain and allow processing in the subsequent layers with feature maps represented in the space domain.

On the other hand, in the approach presented in this article, our idea is to perform all the processing in the frequency domain, without the need to perform inverse transforms, making the training and inference purely in the frequency. Moreover, we are not combining frequency transformation with standard ConvNets, but proposing a new architecture. In this sense, the method will be presented using the Fourier transform, which is calculated only once and can be pre-computed for the input data, greatly reducing the computational cost. Such strategy is especially relevant for problems with

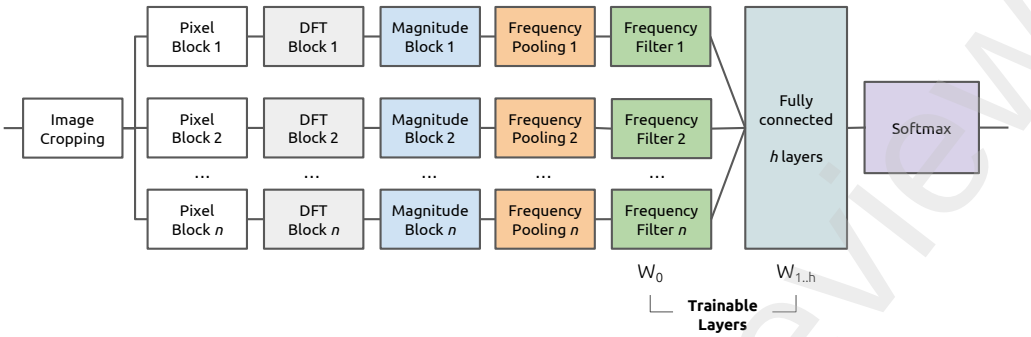

Figure 1: Basic frequency layer architecture for image classification.

appeal to the frequency domain, such as classification of texture images, aerial, microscopic images or for image quality analysis.

3. Proposed Method

3.1. Overview of the Proposed Architecture

The main goal of the proposed method is to explore the frequency domain in order to improve image classification. Figure 1 presents the basic architecture behind the idea exposed in this paper.

The following parameters must be adjusted in the proposed architecture: (1) input image size, (2) number of cropping levels, (3) use of windowing, (4) size and type of frequency pooling, (5) use of the logarithm of magnitude. After performing the frequency filtering, the extracted features are stacked and processed by a fully-connected network. Hence, it is also necessary to choose (6) the number of layers, (7) the number of neurons in each layer, and (8) the activation function.

In addition to the aforementioned aspects, it is also necessary to select the loss function, the training algorithm and its hyperparameters (learning rate, batch size, etc).

Each step and parameter for training the proposed model will be presented in the sequence.

3.2. Image cropping

The first step consists in splitting the input image in several blocks. The purpose of this step is to allow the network to extract both global and local information. When the frequency spectrum is calculated for the entire image,

global features can be extracted; on the other hand, when the image is divided in small blocks, these will provide local features. Examples of global features are the background and main object components, while examples of local features are small textures, corners and borders.

Figure 2 shows the basic cropping procedure. For the sake of clarity, three cropping levels are shown in this figure. Basically, in the first level the whole image is considered. In the second level, the original image is divided in 4 blocks, each one with half of the original height and width. In the third level, each of the four blocks from the previous cropping is divided in four new blocks, thus yielding a total of 16 blocks. This process can be repeated until the chosen maximum number of cropping levels is attained. If a color image is used as input, all the three channels RGB are cropped and processed individually in the next steps.

Figure 2: Basic cropping procedure.

3.3. 2D Discrete Fourier Transform

After the cropping, each block will be transformed to the frequency domain.

In this step, the 2D Discrete Fourier Transform (DFT) [37] is employed. For each image block i, denoted as $f_i(x, y)$, with size $A_i \times B_i$, the DFT is computed according to the following expression:

$$F_i(u, v) = \sum_{x=0}^{A_i-1} \sum_{y=0}^{B_i-1} f_i(x, y) e^{-j2\pi(ux/A_i + vy/B_i)} \tag{1}$$

Figure 3 depicts this step. In this case, considering three cropping levels, 21 DFTs need to be calculated, being one for the entire image, four for the second level and sixteen for the last level.

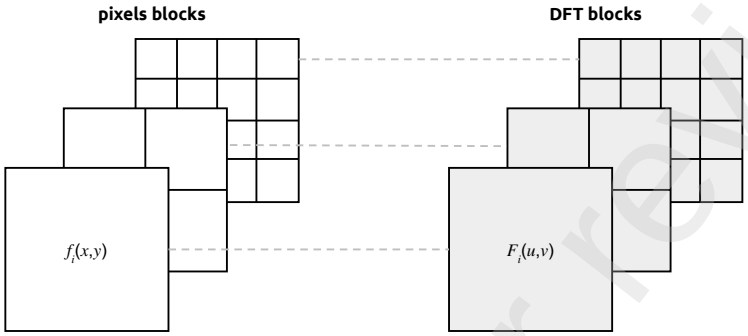

Figure 3: Discrete Fourier Transform for each image block.

Since the DFT yields a matrix of complex numbers and most ANN/DL operations deal with real numbers, the magnitude of the spectrum of each block is obtained, according to Equation (2), and effectively explored by the network. This step is represented in Figure 4.

$$M_i(u,v) = \sqrt{(Real(F_i(u,v))^2 + (Imag(F_i(u,v))^2} \tag{2}$$

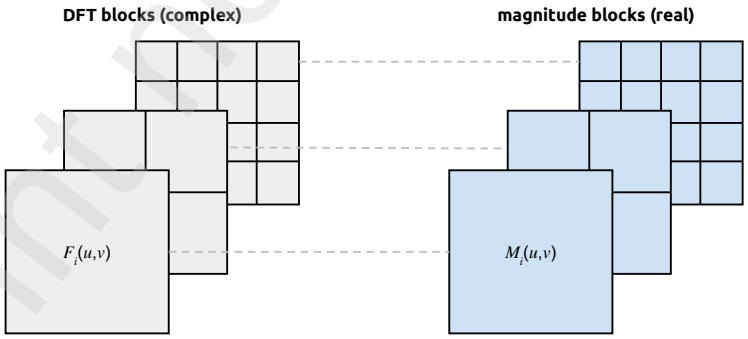

Figure 4: DFT magnitude calculated for each image block.

3.4. Frequency pooling

In the DFT domain, the frequencies radially vary from low, in the center of the spectrum, to the high, in the borders. In this context, frequencies are commonly filtered in a radial way. For example, a band-pass filter can be designed in the shape of a ring, filtering specific frequencies from the image.

In this sense, frequency components associated with close radii can be grouped, if desired. In other words, we may choose to process larger frequency bands instead of single cells sharing the same radius. As shall be described in Section 3.5, the width of the frequency bands directly influences the number of filter weights that need to be tuned by the network. This pooling step is illustrated in Figure 5.

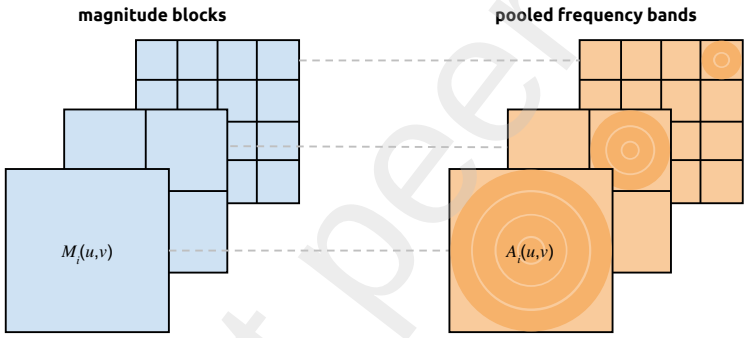

Figure 5: Pooling in frequency to reduce data dimensionality.

Analogously to the case of radial bands, it is also possible to adopt squared frequency groups. In this case, the only difference is that the distance of a particular cell to the center of the image is calculated using Chebyshev distance, instead of Euclidean[38].

The Chebyshev distance grouping can be convenient in this pooling step since it simplifies how the highest frequencies are aggregated, since the maximum circle that can be defined within the magnitude block loses high frequencies in the spectrum borders when using Euclidean distance. The option between radial or squared frequency grouping establishes the pooling type.

Therefore, this operation is useful to reduce the dimensionality for the next steps simplifying and reducing the time spent in the training process.

3.5. Frequency Filtering

After these three steps, the first trainable layer is presented. In this layer, the pooled frequency magnitude will be filtered by weights that will be learned during the training process. In this case, for filtering in the frequency domain, the magnitude will be element-wise multiplied by a new matrix $W_i \in \mathbb{R}^2$, which represents the filter. Figure 6 exhibits this filtering step.

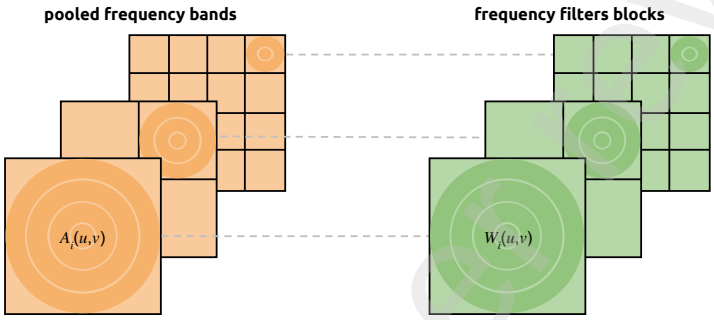

Figure 6: Frequency filtering for each block.

In order to simplify the filtering process, it is possible to define weighted rings or squares, according to the selected pooling type, which will filter each block in the frequency domain. In this case, even though the filter matrix W_i has the same size of the image block, only a few weights need to be adjusted, each representing one radius, as shown in Figure 7. Additionally, as remarked in Section 3.4, if pooling is adopted, the number of weights can be further reduced, as each filter weight is responsible for processing a wider frequency band.

In Figure 7 the frequency filtering will be performed by the element-wise multiplication of the magnitude spectrum (5×5 orange matrix on the left) by the frequency filter (5×5 green matrix on the right). Notice that only three filter weights will be learned on the training step, being: $W_i(1)$, $W_i(2)$ and $W_i(3)$. This process is applied for each block i, multiplying its respective pooled magnitude A_i by the frequency weights W_i, as follows:

$$\mu_i = A_i \odot W_i \tag{3}$$

After this multiplication, a coefficient $C_i(r)$ is extracted from each ring (or square, depending on the pooling type) with radius r and for each block i. Let R be the maximum distance from the center, which is half of the

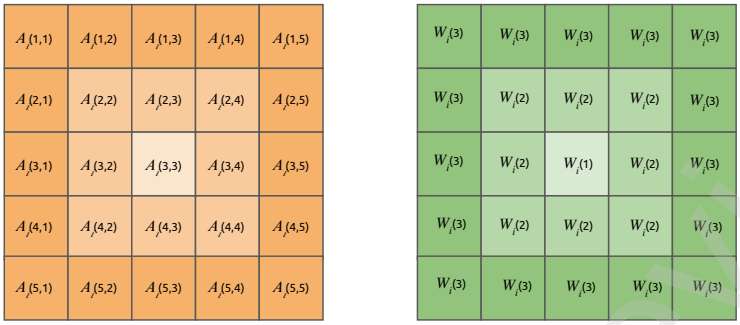

Figure 7: Frequency filtering simplification: cells with the same radial distance share the same frequency weight.

block size, and P the pooling size. Then, the coefficient $C_i(r)$ amounts to the sum of the elements inside the group with radius $[r, r+P]$, as shown in the following expression:

$$C_i(r+1) = \sum_{r'=rP+1}^{(r+1)P+1} \mu_i(r'), r = 0, \ldots, \frac{R}{P} - 1 \tag{4}$$

Using this idea, a substantial reduction in the amount of adjustable coefficients is achieved. For example, given an image of 1024×1024 pixels, there would be 1.048.576 weights to be adapted for the first cropping level. However, using the proposed simplification, only 512 coefficients will be adapted, which makes the training process much faster and feasible.

3.6. Fully connected training

After the frequency filtering step, all the coefficients $C_i(r)$ are stacked and used as inputs for a fully-connected network.

The characteristics of the fully-connected network, such as the number of layers, the number of neurons and the activation functions, can vary as desired for a given application. Since we are considering the task of image classification, the final layer explores the *softmax* function, thus yielding the probability for each class.

It is important to highlight that training the proposed model involves the adaptation of the weights of the fully-connected network, as well as of the filter weights in the frequency filtering stage. Interestingly, the frequency filtering operations, defined in Equations (3) and (4) easily allow the computation of the gradient with respect to the weights. Hence, the training

process of the proposed model backpropagates the error until the frequency layer and updates the weights of the frequency filters.

Since we are processing the magnitude of the spectrum, what is learned by the network is the magnitude spectrum of the filters. Therefore, the weights W_i cannot be negative. A simple way to guarantee this condition consists in truncating to zero the potential negative weights that might arise after the weight update for each mini-batch. Interestingly, this is similar to applying the rectifier function over the filter weights.

By exploring the proposed architecture, with all the ideas and stages already described, interesting results have been obtained. The methodology and experimental results are presented in the next section.

4. Experiments, Results and Discussion

The proposed method was evaluated in three different classification tasks: (1) texture classification, using the Kylberg dataset [39]; (2) cataract detection, using the Kaggle cataract dataset [40]; and (3) retina image quality classification, based on the EyeQ dataset [41]. In all these tasks, there is a strong motivation for exploring the frequency domain, as the involved classes tend to present distinct spectral patterns.

The models were trained on Google Colaboratory using a GPU Tesla T4 with 16GB of memory. All the algorithms are available in a GitHub repository: access here.

4.1. Texture Classification

4.1.1. Kylberg Dataset

The Kylberg texture dataset is composed of images from 28 textures classes, such as rice, stone, lentils, rug and wall. Figure 8 shows one example of each class.

Each image has 576×576 pixels compressed in 8-bit PNG format. Each class has 160 unique images, resulting in 4480 images. Figure 9 shows some examples of the *grass* class, where it is possible to see that the textures vary based on the place where the image was captured. In the case of the *grass* class, different types of plants are present in the image.

The dataset also contains classes from closer categories, such as rices from different species, named *rice 1* and *rice 2*. Figure 10 exhibits one example of these two types of rice.

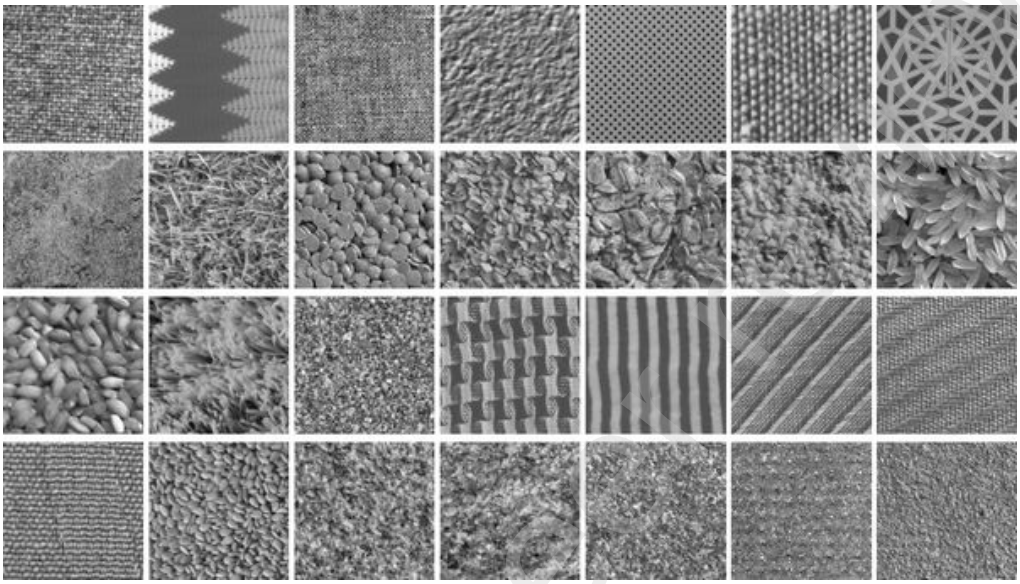

Figure 8: Examples of each class of the Kylberg dataset[39].

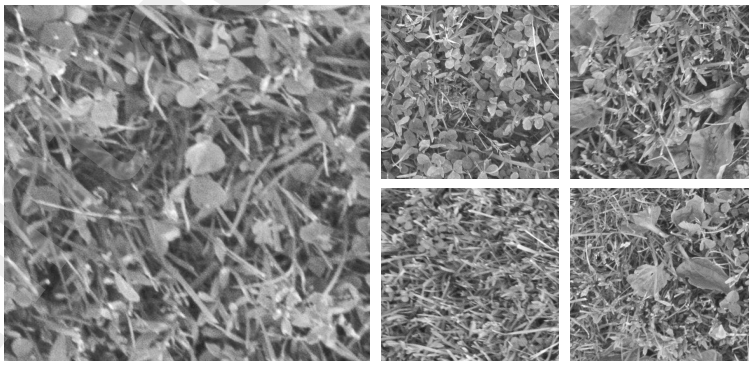

Figure 9: Example of *grass* class from Kylberg texture dataset.

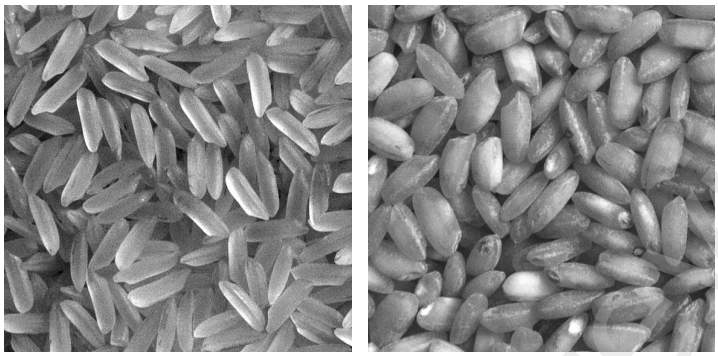

Figure 10: Example of *rice 1* and *rice 2* classes from Kylberg dataset.

4.1.2. Experimental Results (Kylberg Dataset)

Figure 11 displays the architecture employed for this problem.

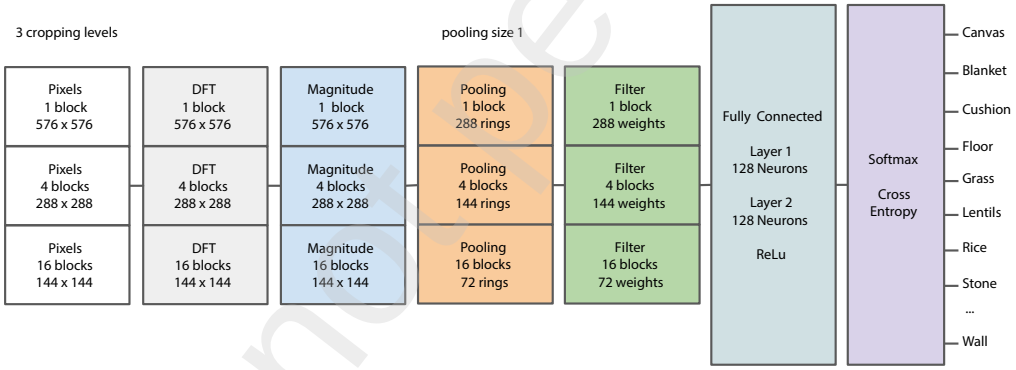

Figure 11: Architecture explored in the case of texture classification (Kylberg dataset).

The training was made with the aid of the Stochastic Gradient Descent (SGD) optimizer and the following parameter values were adopted: learning rate = 0.01, learning rate decay = 0.005, batch size = 10, momentum = 0.8 and epochs = 500. We separated 75% of the available images for the training set, while the remaining 25% of the images were used in the validation set.

The proposed model reached a validation accuracy of 99.73%, demonstrating a good capacity in separating several classes. Figure 12 exhibits two examples of classification errors, in which the network confused the *rug* and *grass* classes. As we can notice, the images are, in fact, quite similar.

Figure 12: Classification error for 28 Kylberg textures: (*rug* and *grass* classes).

It is important to stress that the majority of images from these two classes were correctly classified by the network, which means that this kind of misclassification is an exception, probably due to the internal variations in these two classes.

We also employed some ConvNets in the same scenario in order to analyze whether the proposed model attains a competitive performance. In this case, we used transfer learning by retraining the structure of the ConvNets pretrained on ImageNet dataset [20]. The last layers of the architectures were modified in order to adapt them to the Kylberg texture classification problem. Also, the input image size was adapted to the ConvNets, which requires 224×224 pixel images in most cases and 383×384 pixels in EfficientNetV2S.

Table 1 describes the characteristics (number of parameters and number of layers) of the ConvNets and of the proposed model and presents the obtained classification accuracy in the validation set, as well as the computational time required for the training process. The selected ConvNets were: DenseNet [42], MobileNet [43], Efficient-Net [44], and ResNet [45]. To analyze whether the premise that using more cropping levels leads to a more extensive information extraction, we included results of the proposed method for different numbers of cropping levels.

As we can observe in Table 1, the ConvNets reached a slightly superior accuracy. Nonetheless, the number of adjustable parameters is significanly reduced when we consider the proposed approaches, as well as the corresponding training time. This expressive economy in terms of computational cost comes at the expense of a minor performance reduction, which indicates

Table 1: Comparison between ConvNets and the proposed model for texture classification (Kylberg dataset).

Model	**Parameters**	**Layers**	**Training time**	**Accuracy**
Proposed-3	0.280 M	4	7m 22s	$\mathbf{99,73}\%$
Proposed-2	0.131 M	4	6m 22s	$99,46\%$
Proposed-1	0.057 M	4	6m 06s	$99,02\%$
DenseNet169	12.65 M	170	27m 29s	100%
MobileNetV2	2.26 M	54	11m 53s	$99,82\%$
EfficientNetV2S	20.33 M	171	57m 39s	100%
ResNet50	23.57 M	51	21m 27s	$99,91\%$

that the proposed method can be competitive in this kind of classification task. Additionally, we can notice that as the number of cropping levels is increased, there are more frequency parameters to adjust in the structure, but the accuracy is improved, which suggests that more levels do yield more local and global features that are useful for the classification.

4.2. Retina Experiments

4.2.1. Cataract Dataset

The next experiments involved medical datasets associated with retina images. Initially, we explored the Kaggle Cataract Dataset [40], which is relatively small and contains 300 images of the *normal* class and 100 images of the *cataract* class.

Since we are interested in evaluating the benefits of using frequency-domain information in classification, these two classes are specially interesting to analyze, since this disease reduces the high frequency components in the retina images of cataract patients. Figure 13 presents some samples of this dataset: the two columns on the left present cataract images, whereas normal images are presented in the two columns on the right.

It is possible to observe that the cataract images are cloudy or blurred when compared to the normal eye. This happens since the crystalline becomes opaque in this disease and avoids the light to properly reach the retina.

4.2.2. Experimental Results on Cataract Detection

The challenge in this experiment is to identify whether the patient has cataract or not by processing the retina image. In this case, the 400 images

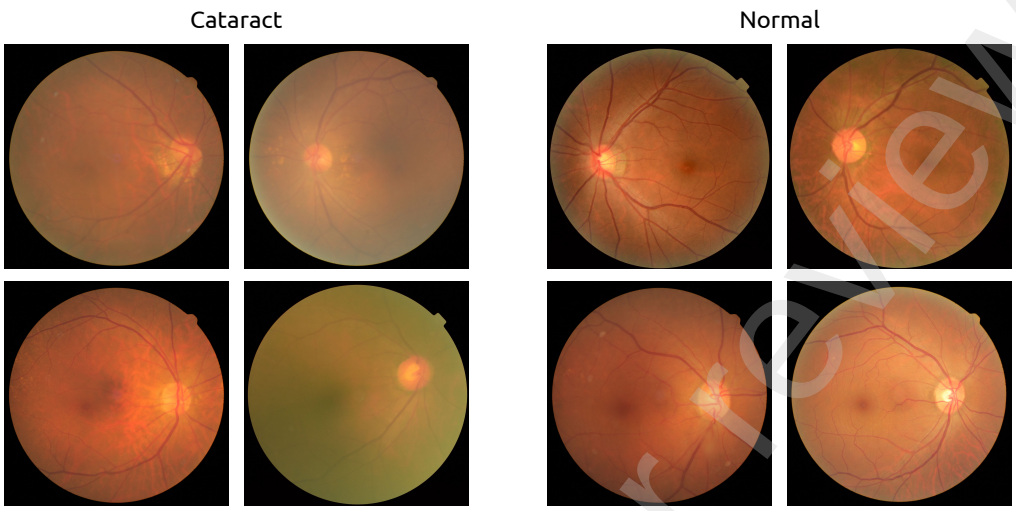

Figure 13: Images from Kaggle cataract dataset.

comprising *normal* and *cataract* classes were separated in training (75%) and validation (25%) sets. Here, each image size was set to 512×512 pixels.

Figure 14 presents the proposed frequency-domain architecture explored in this problem. The SGD optimizer was also employed for this experiment and the training parameters were defined as follows: learning rate = 0.1, learning rate decay = 0.1, batch size = 10, momentum = 0.9 and epochs = 250.

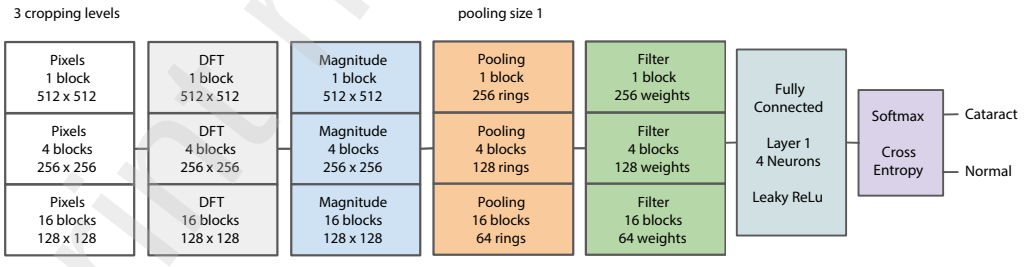

Figure 14: Architecture used to classify *cataract* and *normal* classes.

The same ConvNets were also applied to this task, considering the pre-trained versions on ImageNet. The output layer was replaced by a two-class

softmax to produce the probabilities of *normal* and *cataract* classes and the input image size was also adapted according to the ConvNets requirements. The obtained results are summarized in Table 2.

Table 2: Comparison between ConvNets and the proposed model for cataract detection.

Model	**Parameters**	**Layers**	**Training time**	**Accuracy**
Proposed	8.97 K	2	0m 19s	$\mathbf{93,00}\%$
DenseNet169	12.65 M	170	6m 25s	$90,00\%$
MobileNetV2	2.26 M	54	0m 54s	$91,00\%$
EfficientNetV2S	20.33 M	171	10m 43s	$93,00\%$
ResNet50	23.57 M	51	0m 83s	$88,00\%$

As we can observe, the proposed approach reached the best performance in this scenario, together with the EfficientNetV2S, but with much less parameters and with the fastest training process. In fact, even using transfer learning for the ConvNets, the validation accuracy could not be improved, probably due to the elevated number of parameters in these deep architectures, and to the reduced number of samples in this dataset.

Another aspect that may have contributed to the success of the proposed approach is that the Fourier spectra of *cataract* and *normal* images usually are very different, since normal images have much more higher frequency components than cataract images. So, by carefully designing the frequency filters, it can effectively explore these characteristics of the images to improve the classification.

Another interesting aspect to analyze refers to the number of parameters in these architectures. The proposed method constitutes a very simple and lightweight model, with only 8.9 thousand parameters, while the lighter ConvNet compared has more than 2.2 million. Still, the performance of the proposed method was much better considering its generalizability against its number of trainable parameters, demonstrating that it can be an attractive solution for some classification problems.

In the next section a large retina dataset will be evaluated, but from the perspective of classifying the quality of the images.

4.2.3. EyePACS and EyeQ Dataset

Assessing the retina image quality is an essential factor in order to allow the correct diagnosis. Many images are captured with differences in illumination, focus, and proper alignment with the eye fundus. The retina image

quality index is also important to instruct the medical operator to repeat the image capture if it is out of focus, for example, or even can be used as a filter for machine learning algorithms that automatically analyze retina images in order to detect diseases like DR and glaucoma. An interesting dataset in this field is based on EyePACS project[46].

The EyePACS project provides an excellent dataset for DR detection and it is available on Kaggle platform [47]. It provides about 90 thousand retina images from 5 different classes, each one identifying a DR level: *normal*, *mild*, *moderate*, *severe* and *proliferative*.

This dataset is big and widely employed for DR classification, but it is a real-world dataset, which means that there is noise on images and labels. In this sense, the images contain artifacts being, for example, under or over-exposed and also out of focus. This fact is quite pertinent for the current experiment since we are interested in separating good from bad quality images.

The EyeQ dataset[41] provides a subset of the EyePACS, dividing 28,792 images in three classes according to the image quality: *good*, *usable* and *reject*. Figure 15 exhibits some examples from this dataset.

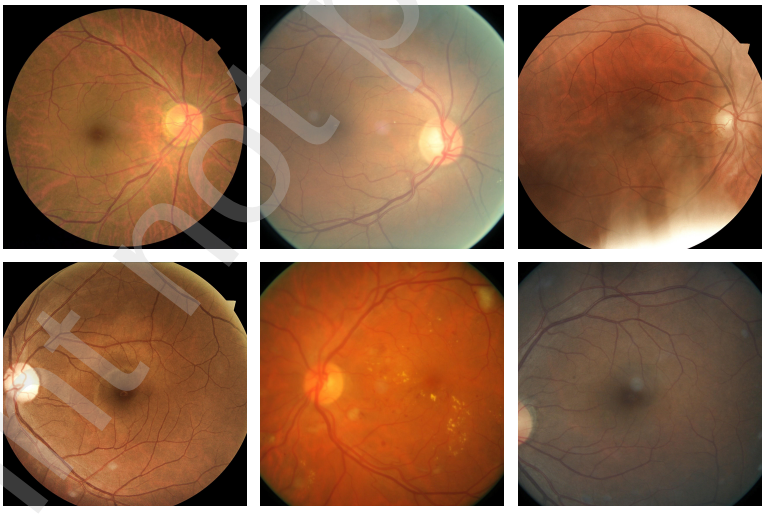

Figure 15: EyeQ dataset for retina quality evaluation. Classes: *good* (left), *usable* (center) and *reject* (right).

4.2.4. Experimental Results on Retina Quality Evaluation

In this experiment, two classes of the EyeQ dataset were explored: *good* and *reject*. The idea is that by automatically recognizing the good and the bad images, it is possible to instruct an operator to repeat the acquisition whenever necessary, and also to help him/her during the examination. Considering these two classes, 22,358 images with 512×512 pixels were used for training and evaluating the models, according to the following division [41]: Training set – 8,347 images of *good* category and 2,320 images of *reject* category; validation set – 8,471 images of *good* category and 3,220 images of *reject* category.

Using this dataset, we trained the proposed architecture exhibited in Figure 16 with the aid of the Adam optimizer and the following parameters were adopted: learning rate = 0.001, batch size = 32 and epochs = 700.

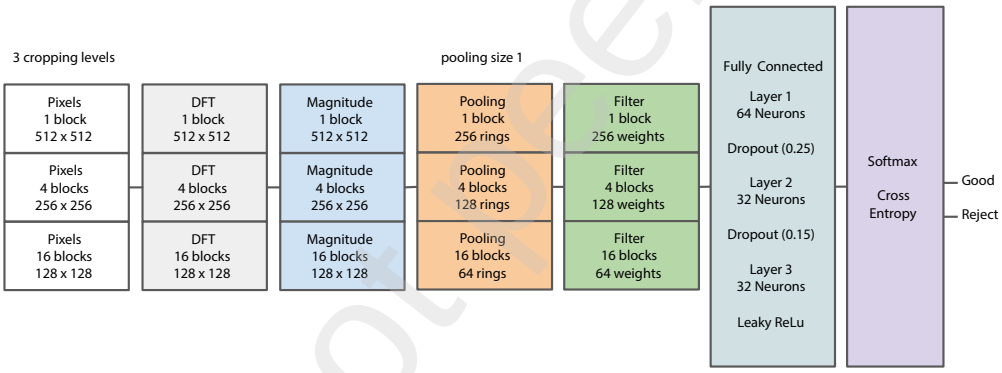

Figure 16: Architecture used to classify retina quality.

For the sake of comparison, DenseNet [42], MobileNet [43], EfficientNet [44] and ResNet [45] recent versions were also trained. Similarly to the previous experiments, these models were pre-trained on ImageNet dataset and transfer learning was applied in order to adapt them to the EyeQ images; also, the input image size was modified according to the ConvNets requirements. Here, two neurons on the output layer were considered. The details concerning the architectures and the corresponding performances are presented in Table 3.

By analyzing the results, it is possible to observe that EfficientNetV2S achieved the best accuracy, but, requiring 20 million parameters to achieve this score. DenseNet169, MobileNetV2 and ResNet50 follow with accuracies

Table 3: Comparison between ConvNets and the proposed model for retina quality classification.

Model	Parameters	Layers	Training time	Accuracy
Proposed	0.353 M	4	10min 95s	**98, 39**%
DenseNet169	12.65 M	170	29min 51s	98, 68%
MobileNetV2	2.26 M	54	12min 58s	98, 50%
EfficientNetV2S	20.33 M	171	476min 22s	98, 82%
ResNet50	23.57 M	51	22min 36s	98, 65%

close to 98.6%. Interestingly, the proposed frequency-based approach is competitive, achieving a classification accuracy just 0.48% below the best result, even with a relatively simple and shallow architecture, employing only four dense layers and a total of 353 thousand parameters.

It is essential to remark that the chosen ConvNets were pre-trained on the ImageNet dataset, thus leveraging their performances by resorting to parameters tuned on such a large dataset, whereas the proposed model was trained from scratch using solely the available samples for each target task.

Hence, in order to establish a more balanced comparison with the proposed method, and following the methodology adopted in related works of this area [48, 49, 50], all architectures were trained from scratch, i.e., starting from a random initial condition, and facing the same challenge: learning to identify the classes given only the available data.

4.2.5. Comparative Analysis without Transfer Learning

Now, the chosen ConvNets were retrained and applied to the selected three tasks (texture classification, cataract detection and retina quality classification), but considering random initial weights (i.e., without transfer learning). Table 4 presents the corresponding results for the three scenarios. We reproduce the results of the proposed method to facilitate the comparison.

By comparing the accuracy values shown in Table 4 with those reported in Sections 4.1 and 4.2, it is possible to verify that the use of pre-trained layers on ImageNet was decisive for the performance of the ConvNets. In fact, when the network weights are randomly initialized and ConvNets have to learn from scratch – which is a fairer condition to the comparison with the proposed method – they are surpassed by our frequency-based approach in all the considered datasets.

This aspect is interesting because it highlights the relevance of working in

Table 4: Summary of the results obtained by the ConvNets and the proposed method for all the scenarios. Now, all models are adapted given only the target datasets and from random initializations.

Dataset	Model	Parameters	Layers	Training time	Accuracy
Kylberg	Proposed	0.280 M	4	7m 22s	**99, 73%**
	DenseNet	12.5 M	170	75m 36s	100%
	MobileNet	2.26 M	54	6min 8s	4, 29%
	EfficientNetV2S	20.33 M	171	60m 14s	99, 46%
	ResNet50	23.57 M	51	25m 9s	97, 77%
Cataract	Proposed	8.97 K	2	0m 19s	**93, 00%**
	DenseNet	12.5 M	170	5m 13s	84, 00%
	MobileNetV2	2.26 M	54	2m 2s	78, 00%
	EfficientNetV2S	20.33 M	171	8m 11s	88, 00%
	ResNet50	23.57 M	51	2m 25s	83, 00%
EyeQ	Proposed	0.353 M	4	10min 95s	**98, 39%**
	DenseNet	12.5 M	170	80min43s	97, 95%
	MobileNetV2	2.26 M	54	72min 55s	98, 27%
	EfficientNetV2S	20.33 M	171	82min 54s	96, 54%
	ResNet50	23.57 M	51	96min23s	97, 36%

the frequency domain in these types of problems, and because the proposed model is relatively simple and with much faster training.

Also, it is important to mention that MobileNetV2 was not capable of classifying the 28 Kylberg classes without the use of transfer learning, so its results were not considered on Table 4.

Another important remark is that the training time associated with the proposed approach can be improved by using more robust parallel processing and hardware, which can also allow training deeper frequency networks with more parameters, contributing to improving the accuracy of the classification. In addition, regularization and normalization techniques can be applied, which may bring additional benefits to the performance of the model. Therefore, the proposed approach can be considered a promising alternative to the image classification task, especially in cases where the spectral data analysis is known to be relevant.

5. Conclusion

Frequency domain brings new possibilities to improve results in machine learning applications. Deep learning has achieved excellent results for many

problems, but there is always room for improvement. Problems that have a great frequency dependency, like the ones related to image quality classification, may be suitably tackled by using simpler models based on the spectrum components.

Considering this, in this paper, a new method for training frequency domain layers for image classification was presented. The method applies the idea of iteratively splitting the image in small blocks in order to extract global and local information from each part of the image. Each block is then transformed to the frequency domain by using the Discrete Fourier Transform. Using the magnitude of each block, frequency filters were trained based on the back-propagation algorithm.

As presented in Section 2, there are some works in the literature that explore the frequency domain along with deep learning techniques. However, most of them still keep part of the processing in the spatial domain, so that several inverse transformations between frequency and space domains are applied within these models. On the other hand, the approach presented in this work performs all the processing in the frequency domain, even in the deepest and fully connected layers. This makes the network to process feature maps in the frequency domain, which ends up simplifying both training and inference processes. Within our knowledge and based on our research, this is the first work that uses the Fourier transform of the RGB channels of the input images and makes the processing, learning and inference entirely in the frequency domain.

The proposed method was analyzed in different classification scenarios, in which exploring the frequency-domain information seems to be appealing. The first experiment involved image textures belonging to 28 classes from the Kylberg dataset. The attained results were very satisfactory, since the proposed model reached a competitive performance in terms of accuracy, using a very simple architecture and 8 times less parameters than the lightest ConvNet.

In the sequence, a new experiment involved a cataract dataset available on the Kaggle platform. It is a small dataset, but it allows to separate the cataract eyes from the normal ones, through retina images. In this case, compared to DenseNet169, MobileNetV2 and ResNet50, the proposed model achieved superior performance even though it requires only 4 neurons on the intermediate dense layer and 8 thousand adjustable parameters. EfficientNetV2S was the only ConvNet that achieve the same validation accuracy; however, it required 20 million adjustable parameters, 2500× more than Fre-

qNet. In general, the good result achieved by the frequency-based model may be associated with the observation that retina images from normal and cataract patients can be more easily distinguished in the frequency domain, since the images of the cataract are blurred and foggy due to the opaque crystalline caused by the disease.

Afterwards, the performance for another retina dataset was evaluated, but with the objective of distinguishing good and bad images, considering the level of image quality. In this experiment, the dataset was larger and EfficientNetV2S achieved the best performance with 98.82% accuracy, while the proposed method with three cropping levels reached 98.34%. However, the images reshape to 384×384 required by EfficientNetV2S coupled with limited memory capacity turned the training a lot slow when compared to the other ConvNets which deal with an image size of 224×224. Nevertheless, the proposed model achieved a very close and competitive result, even with much simpler and shallower architecture than the state-of-the-art ConvNets.

It should be noted that, in the experiments with these three distinct databases, all ConvNets were initialized with the weights obtained after training on ImageNet dataset, which certainly gave an advantage to them, because they already have extractors with very optimized image features. The proposed model, on the other hand, was trained from scratch.

Thus, a new experiment was conducted, in which the ConvNets were retrained, but starting from randomly initialized weights. In this case, the proposed model obtained the best performances in the considered datasets, proving its potential for classifying images in scenarios where there is an appeal to the frequency domain.

So, the main contributions of this work correspond to a novel architecture exploring the frequency domain, whose parameters are the frequency filter weights, that yielded competitive results when compared with DNNs architectures for some problems, but being much simpler and lightweight. In this sense, our approach can be attractive for mobile or real-time applications, since it has less parameters but with a satisfactory result, allowing to reduce, for example, the battery consumption when running in a mobile device or even allowing to process video, detecting objects or scenes in real-time.

The same idea proposed in this work can also be applied in other fields, such as audio processing. In this context, it is probably advantageous to let the model learn which are the most discriminative frequencies to distinguish between the classes (e.g., male and female speaker), instead of using a generic feature extraction technique, like MFCC.

Finally, this work also raises the discussion whether the frequency domain can be combined with spatial information in order to improve the image classification. It is clear that some tasks may be more appropriately addressed in the frequency domain. So, exploring standard DNNs with frequency features may help to improve even more the results, being a good field for future research.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001 and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). The authors also greatfully acknowldge the financial and mentoring support given by DeepMind Technologies Ltd (5629).

References

[1] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, " O'Reilly Media, Inc.", 2022.

[2] O. M. Molchanov, K. D. Launey, A. Mercenne, G. H. Sargsyan, T. Dytrych, J. P. Draayer, Machine learning approach to pattern recognition in nuclear dynamics from the ab initio symmetry-adapted no-core shell model, Phys. Rev. C 105 (2022) 034306. doi:10.1103/PhysRevC.105.034306.
URL https://link.aps.org/doi/10.1103/PhysRevC.105.034306

[3] X. Xu, W. Qiu, W. Li, X. Liu, Z. Zhang, X. Li, D. Luo, Associations between street-view perceptions and housing prices: Subjective vs. objective measures using computer vision and machine learning techniques, Remote Sensing 14 (4) (2022) 891.

[4] G. Deshpande, A. Batliner, B. W. Schuller, Ai-based human audio processing for covid-19: A comprehensive overview, Pattern recognition 122 (2022) 108289.

[5] C. Caucheteux, J.-R. King, Brains and algorithms partially converge in natural language processing, Communications biology 5 (1) (2022) 1–10.

[6] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al., Grandmaster level in starcraft ii using multi-agent reinforcement learning, Nature 575 (7782) (2019) 350–354. doi:10.1038/s41586-019-1724-z.

[7] A. R. Pathak, M. Pandey, S. Rautaray, Application of deep learning for object detection, Procedia computer science 132 (2018) 1706–1717.

[8] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, B. Yu, Recent advances in convolutional neural network acceleration, Neurocomputing 323 (2019) 37–51.

[9] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel, Handwritten digit recognition with a back-propagation network, Advances in neural information processing systems 2 (1989).

[10] M. Dentino, J. McCool, B. Widrow, Adaptive filtering in the frequency domain, Proceedings of the IEEE 66 (12) (1978) 1658–1659. doi:10.1109/PROC.1978.11177.

[11] M. Mathieu, M. Henaff, Y. LeCun, Fast training of convolutional networks through ffts (2013). doi:10.48550/ARXIV.1312.5851.
URL https://arxiv.org/abs/1312.5851

[12] J. A. Stuchi, M. A. Angeloni, R. F. Pereira, L. Boccato, G. Folego, P. V. S. Prado, R. R. F. Attux, Improving image classification with frequency domain layers for feature extraction, in: 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP), 2017, pp. 1–6.

[13] V. Srinivasan, C. Eswaran, N. Sriraam, Artificial neural network based epileptic detection using time-domain and frequency-domain features, Journal of Medical Systems 29 (6) (2005) 647–660.

[14] A. Malliani, M. Pagani, F. Lombardi, S. Cerutti, Cardiovascular neural regulation explored in the frequency domain., Circulation 84 (2) (1991) 482–492.

[15] K. Wang, N. An, B. N. Li, Y. Zhang, L. Li, Speech emotion recognition using fourier parameters, IEEE Transactions on affective computing 6 (1) (2015) 69–75.

[16] F. U. Dowla, S. R. Taylor, R. W. Anderson, Seismic discrimination with artificial neural networks: preliminary results with regional spectral data, Bulletin of the Seismological Society of America 80 (5) (1990) 1346–1373.

[17] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Acoustics, Speech and Signal Processing (ICAASP), 2013 IEEE International conference on, IEEE, 2013, pp. 6645–6649.

[18] D. A. Silva, J. A. Stuchi, R. P. V. Violato, L. G. D. Cuozzo, Exploring convolutional neural networks for voice activity detection, in: Cognitive Technologies, Springer, 2017, pp. 37–47.

[19] T. Brosch, R. Tam, Efficient training of convolutional deep belief networks in the frequency domain for application to high-resolution 2d and 3d images, Neural computation 27 (1) (2015) 211–227.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[21] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, R. L. Buckner, Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults, Journal of cognitive neuroscience 19 (9) (2007) 1498–1507.

[22] O. Rippel, J. Snoek, R. P. Adams, Spectral representations for convolutional neural networks, in: Advances in neural information processing systems, 2015, pp. 2449–2457.

[23] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

[24] Y. Wang, C. Xu, S. You, D. Tao, C. Xu, Cnnpack: Packing convolutional neural networks in the frequency domain, in: Advances in neural information processing systems, 2016, pp. 253–261.

[25] H. Pratt, B. Williams, F. Coenen, Y. Zheng, Fcnn: Fourier convolutional neural networks, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2017, pp. 786–798.

[26] Y. LeCun, C. Cortes, C. J. Burges, The mnist database of handwritten digits (1998).

[27] Z. Liu, J. Xu, X. Peng, R. Xiong, Frequency-domain dynamic pruning for convolutional neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 1043–1053.

[28] S. O. Ayat, M. Khalil-Hani, A. A.-H. Ab Rahman, H. Abdellatef, Spectral-based convolutional neural network without multiple spatial-frequency domain switchings, Neurocomputing 364 (2019) 152–167.

[29] A. L. Cambridge, The database of faces, Site: www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase (2002).

[30] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[31] J. Lin, L. Ma, Y. Yao, A fourier domain training framework for convolutional neural networks based on the fourier domain pyramid pooling method and fourier domain exponential linear unit, IEEE Access 7 (2019) 116612–116631.

[32] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, F. Ren, Learning in the frequency domain, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1740–1749.

[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[34] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.

[35] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European conference on computer vision, Springer, 2014, pp. 740–755.

[37] R. C. Gonzalez, R. E. Woods, S. L. Eddins, Digital Image Processing Using MATLAB, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[38] A. R. Webb, Statistical pattern recognition, John Wiley & Sons, 2003.

[39] G. Kylberg, The kylberg texture dataset v. 1.0, External report (Blue series) 35, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden (September 2011).
URL http://www.cb.uu.se/~gustaf/texture/

[40] Y. Chen, Kaggle - cataract dataset, Site: www.kaggle.com/jr2ngb/cataractdataset (08 2019).

[41] H. Fu, B. Wang, J. Shen, S. Cui, Y. Xu, J. Liu, L. Shao, Evaluation of retinal image quality assessment networks in different color-spaces, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019, pp. 48–56.

[42] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, Densely connected convolutional networks (2016). doi:10.48550/ARXIV.1608.06993.
URL https://arxiv.org/abs/1608.06993

[43] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861 (2017).

[44] M. Tan, Q. V. Le, Efficientnet: Rethinking model scaling for convolutional neural networks (2019). doi:10.48550/ARXIV.1905.11946.
URL https://arxiv.org/abs/1905.11946

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition (2015). doi:10.48550/ARXIV.1512.03385.
URL https://arxiv.org/abs/1512.03385

[46] J. Cuadros, G. Bresnick, Eyepacs: an adaptable telemedicine system for diabetic retinopathy screening, Journal of diabetes science and technology 3 (3) (2009) 509–516.

[47] Eyepacs, Kaggle - diabetic retinopathy detection, Site: www.kaggle.com/c/diabetic-retinopathy-detection (02 2015).

[48] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, J. Liang, Convolutional neural networks for medical image analysis: Full training or fine tuning?, IEEE transactions on medical imaging 35 (5) (2016) 1299–1312.

[49] J. Pons, X. Serra, Randomly weighted cnns for (music) audio classification, in: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2019, pp. 336–340.

[50] A. V. Ikechukwu, S. Murali, R. Deepu, R. Shivamurthy, Resnet-50 vs vgg-19 vs training from scratch: a comparative analysis of the segmentation and classification of pneumonia from chest x-ray images, Global Transitions Proceedings 2 (2) (2021) 375–381.