computers

MDPI

Article

Identifying Bias in Deep Neural Networks Using Image Transforms

Sai Teja Erukude, Akhil Joshi and Lior Shamir *

Department of Computer Science, Kansas State University, Manhattan, KS 66502, USA; erukude.saiteja@gmail.com (S.T.E.); akhil.joshi08@gmail.com (A.J.)
* Correspondence: lshamir@mtu.edu

Abstract: CNNs have become one of the most commonly used computational tools in the past two decades. One of the primary downsides of CNNs is that they work as a "black box", where the user cannot necessarily know how the image data are analyzed, and therefore needs to rely on empirical evaluation to test the efficacy of a trained CNN. This can lead to hidden biases that affect the performance evaluation of neural networks, but are difficult to identify. Here we discuss examples of such hidden biases in common and widely used benchmark datasets, and propose techniques for identifying dataset biases that can affect the standard performance evaluation metrics. One effective approach to identify dataset bias is to perform image classification by using merely blank background parts of the original images. However, in some situations, a blank background in the images is not available, making it more difficult to separate foreground or contextual information from the bias. To overcome this, we propose a method to identify dataset bias without the need to crop background information from the images. The method is based on applying several image transforms to the original images, including Fourier transform, wavelet transforms, median filter, and their combinations. These transforms are applied to recover background bias information that CNNs use to classify images. These transformations affect the contextual visual information in a different manner than it affects the systemic background bias. Therefore, the method can distinguish between contextual information and the bias, and can reveal the presence of background bias even without the need to separate sub-image parts from the blank background of the original images. The code used in the experiments is publicly available.

Keywords: bias; convolutional neural networks; machine learning; experimental design

check for updates

Citation: Erukude, S.T.; Joshi, A.; Shamir, L. Identifying Bias in Deep Neural Networks Using Image Transforms. *Computers* **2024**, *13*, 341. https://doi.org/10.3390/computers13120341

Academic Editor: Robertas Damaševičius

Received: 19 November 2024
Revised: 9 December 2024
Accepted: 13 December 2024
Published: 15 December 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the past decade, CNNs have revolutionized the field of computer vision due to their unprecedented performance capabilities with regard to image analysis, combined with their ease-of-use due to the availability of libraries. These networks use automatic extraction and selection of data-driven abstract features, which means a substantial reduction in the need for manual feature engineering. CNNs are currently widely used in a variety of applications, including object detection, recognition, medical imaging, and many more.

With these substantial advantages, CNNs also have notable limitations. For example, training CNNs is computationally expensive, normally involving the need for advanced GPUs or computing clusters for execution. Training CNNs often requires a large number of samples [1], and overfitting is a common situation when using CNNs [2–5]. The need for large training sets leads to the common use of data augmentation, which can in itself affect the bias in a controlled or uncontrolled manner [6,7]. Changing the size of the input images can also change the performance of a CNN [8].

In addition to these limitations, one of the primary downsides of CNNs is their "black box" nature. While CNNs are trained with data samples, the process of updating their weights through the training remains opaque. The rules that define how classifications

are made are convoluted and unintuitive, and it has been proven difficult to interpret the reason behind decisions made by a deep neural network [9]. The definition of what a CNN really "learns" from the data thus becomes equally difficult to establish. In other words, lack of interpretability simply means that CNNs should be used with caution as their classification processes are not necessarily clear to the user [10,11].

Ideally, a CNN that can achieve high accuracy in classification on benchmark datasets should perform well also in real-world settings. On the other hand, empirical studies [9,12,13] show that many popular datasets are biased, and possibly not representative of a model's performance concerning real-world image recognition. That shows that CNNs trained on benchmark datasets might be driven by information that is not limited to the information related to the task they aim at solving.

Tools that are used to identify biases or irrelevant information that drives the classification accuracy include saliency maps [14–16], which can be used to visualize the signal through the spacial domain. This allows for inspecting and identifying sources of signal that might not be relevant to the computer vision problem at hand [17,18]. The domain of adversarial neural networks has also been studied in the context of irrelevant information that can skew the way CNNs work, yet with no direct link to the relevant information [19,20]. Some solutions have also been proposed to detect adversarial samples [21,22]. The need of deep neural networks for large datasets has led to the practice of data augmentation. Data augmentation has been used to correct for a variety of biases in datasets, especially biases related to the distribution of the data and fairness of the analysis [23–26].

As the use of CNNs in various fields continues to grow, it is imperative to address these limitations, which also include the risk of classification bias. This reinforces the need to develop methods that can assist researchers who develop CNNs, and allow them to reduce the risk of unknown or unexpected biases.

2. Classification Bias in Benchmark Datasets

The term "bias" in the context of benchmark datasets for image classification has been used to describe numerous situations. These may include unbalanced distribution of the samples in the classes, biases in the labeling of the images, biased selection of data, and more [27]. Unbalanced distribution of the samples in datasets is a well-known bias driven by an unequal number of samples in different classes. Machine learning can be biased by the number of samples a machine learning model is trained on, making more populated classes become more frequent in the classification outcomes. This leads to some classes with a higher error rate than other classes, consequently leading to problems pertaining to fairness and discrimination associated with machine learning algorithms. One of the common solutions to the problem is balancing the classes through data augmentation [23–26].

Labeling bias is a situation of incorrect or inconsistent labeling. It is related to contextual information bias, in which elements that are not the target objects are present in one class more frequently than in other classes, and can therefore be used by a classifier to identify the class. Such biases can be detected by manual or automatic inspection of the contextual information, and also by comparing the performance when training with one dataset and testing with another dataset [27]. For instance, if the training and testing with two different datasets leads to weaker classification accuracy compared to training and testing with the same dataset, it can be considered an indication of bias [27].

While these biases can be identified, one of the more challenging biases is driven by information that is not part of the visible visual content of the images, yet its presence in the images allows machine learning algorithms to classify the images correctly. Because the source of such bias is not necessarily known, identifying it becomes more challenging, which can deceive even experienced researchers.

Because it is difficult to identify, such classification bias is present in many widely used benchmark datasets [9,12,13,28–31]. For instance, Model and Shamir [29] conducted an in-depth study comparing dataset bias across various traditional object recognition benchmarks, including COIL-20, COIL-100, NEC Animals, Caltech 101, etc. They em-

ployed a method that involved isolating a small, seemingly blank background area from each image—an area that does not contain any information about the object of interest. The findings revealed that all datasets that were tested still achieved classification accuracy above chance levels using these small sub-images, even though the sub-images lacked any visually interpretable information. This technique was used to detect dataset bias in single-object recognition datasets, and to compare the extent of bias across different datasets [29].

A similar observation was made with common face recognition datasets [28]. Experiments showed that by using just a small part of the background that does not include any part of the face or hair, the algorithms could still identify the faces with accuracy far higher than mere chance [28].

The impact of bias is substantially stronger in datasets of objects imaged in controlled environments [13,29]. Such datasets include commonly used benchmarks such as COIL-20, COIL-100, and NEC Animals. Datasets prepared from natural images collected from online sources include ImageNet [32], among others. The fact that these datasets achieved classification accuracy well above mere chance using only seemingly uninformative background areas that seem identical to each other when observed with the naked human eye indicates the presence of artifacts or noise that may influence the classification. This potentially allows machine vision algorithms to classify images without properly identifying the objects, and therefore without solving the machine vision problem they intend to solve.

Majeed et al. [30] encountered similar challenges when they investigated the applicability of CNNs for detecting COVID-19 in chest X-ray images. Their study involved using 12 established CNN architectures in transfer learning mode across three publicly available chest X-ray databases. Additionally, they introduced a custom shallow CNN model, trained from scratch. For their experiments, chest X-ray images were input into CNN models without any pre-processing, mirroring how other studies have used these images.

To better understand how these CNNs made their predictions, they conducted a qualitative analysis using a method called Class Activation Mappings (CAM) [33]. CAMs allow researchers to generate heatmaps of the most discriminating regions and visualize which parts of the input image were most influential in the network's decision-making process. By mapping the activation from the CNN back to the original image, one can highlight class-specific regions that contributed most significantly to the model's classification, providing insights into which part of the image was the most relevant for detecting COVID-19.

Figure 1 illustrates the parts of the images that the CNN models focused on when making predictions. As the figure shows, only in a few cases the models concentrated on the frontal chest region (i.e., the lung area), which is where typically signs of COVID-19 and other infections can be noticed. Instead, these models often focused on areas outside the chest's frontal view, and sometimes outside of the entire body. That can be seen in the first column of row (b) and the third and fourth columns of row (e) in Figure 1. Additionally, there are noticeable overlaps between the CAM hot spots and textual elements in the images, particularly in the first column of row (b), and the first, third, and fourth columns of row (e).

This is a notable example of how almost all CNN architectures used areas outside the region of interest (ROIs) to make the final classification predictions. This observation leads to the conclusion that these images contain information that is not necessarily related to the medical condition being studied, yet can be used to classify between the images. The presence of such information can lead to biased classification accuracy such that the accuracy observed with the benchmark dataset is different from the classification accuracy that the CNN architecture can achieve in real-world settings.

The limitation of the method for the detection of biases is that the images cannot be registered in all cases in a manner that allows a map that shows consistent signal from different parts of the images. Another limitation is that in some cases the entire image covers an area from which signal is expected. In such cases, parts of the images that are known to be irrelevant cannot be inspected separately, and therefore the bias cannot be

identified. When no parts that are irrelevant for the classification are expected, it is difficult to know whether bias exists, as all parts of the image are expected to be informative.

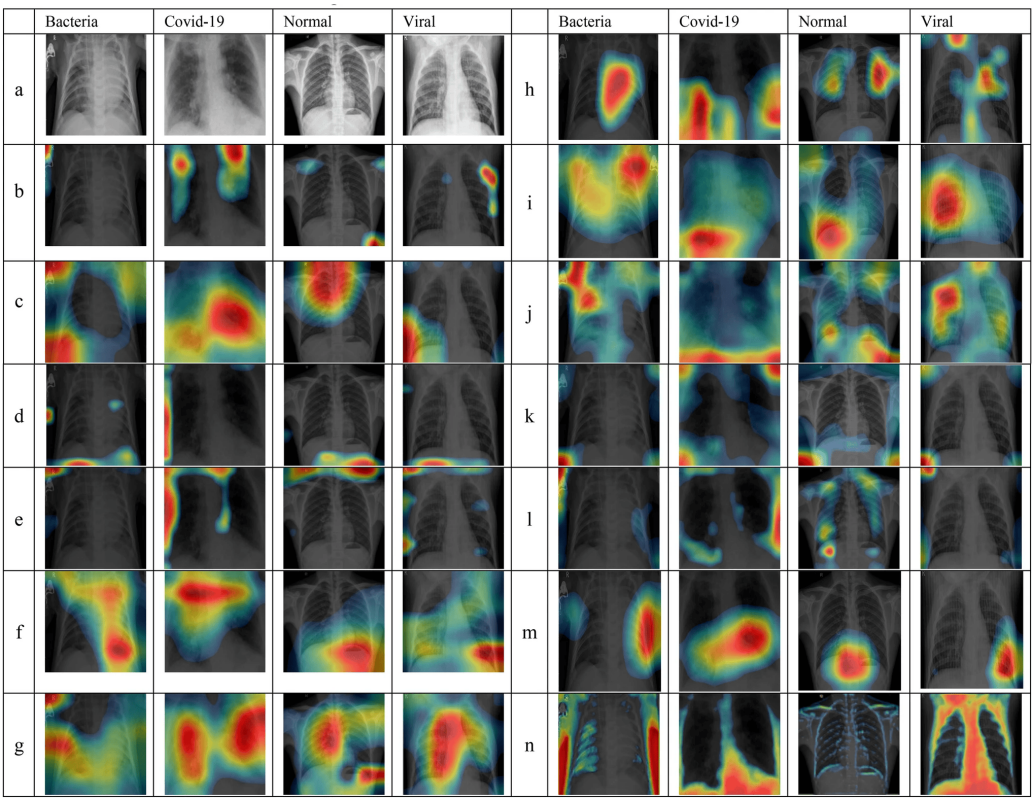


Figure 1. X-rays images classified correctly by CNNs. (**a**) Original X-ray, (**b**) AlexNet, (**c**) GoogleNet, (**d**) VGG16, (**e**) VGG19, (**f**) ResNet18, (**g**) ResNet50, (**h**) ResNet101, (**i**) Inception V3, (**j**) InceptionResNet, (**k**) DenseNet201, (**l**) SqueezeNet, (**m**) Xception, and (**n**) CNN-X [30].

A related study [13] highlighted significant bias-related issues in common biomedical, face recognition, and object recognition benchmark datasets. A primary concern is the tendency of CNNs to learn from background information rather than the relevant visual content of the images. The experiments follow common practices to test whether CNNs can classify these image datasets, but instead of using the entire images, merely parts of the images that did not have any content of the object of interest were used. An example is shown in Figure 2.

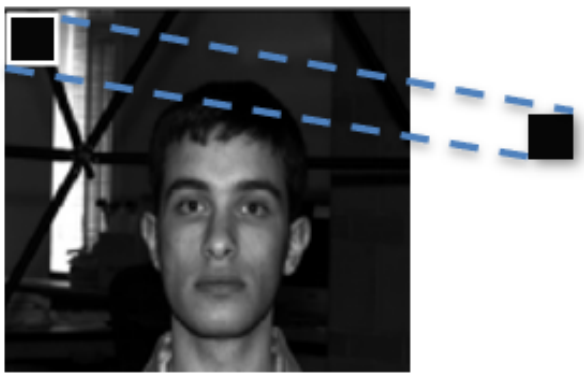


Figure 2. A 20 × 20 cropped background segment of a blank background part of the original image. This was carried out for all images in the dataset to create a new dataset of blank sub-images, as shown in Figure 3. When using just these seemingly blank parts of the images, the classification accuracy of numerous datasets was far higher than mere chance accuracy.

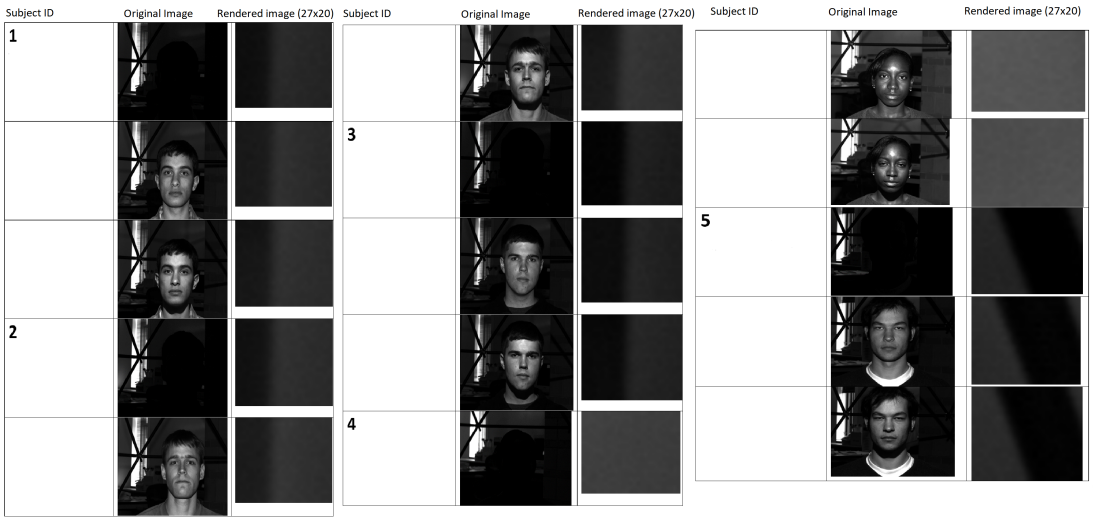


Figure 3. Original images from Yale Faces B and the 20 × 20 portion of the top left corner separated from each of the original images. The classification accuracy of the CNN was far higher than mere chance, showing the CNN does not necessarily need to recognize the face in order to classify the images correctly.

For instance, Figure 3 depicts the original Yale Faces B dataset alongside the newly generated 20 × 20 cropped dataset. The 20 × 20 area of the image was selected based on previous experiments [13,28,29], with the motivation of including just seemingly blank background parts of the images. Foreground parts of the images that contained information relevant to the image classification problem were not included in the sub-images.

When a CNN was trained on this cropped dataset, it achieved a surprisingly high classification accuracy of ∼87.8%, which is far higher than the expected random accuracy of about ∼3.57% [13]. Moreover, in the Kvasir dataset, which contains images of endoscopic examinations, CNNs were able to classify blank 20 × 20 pixel sub-images as shown in Figure 4. In this case, the dataset made of small parts of the background provided classification accuracy of ∼30.75%. This accuracy is also significantly higher than the expected random chance accuracy of ∼12.5%. This raises questions about the reliability of models trained on such datasets, as they may exploit biases introduced by the image acquisition process rather than genuinely discerning clinical conditions.

Experiments using various biomedical datasets, including COVID-CT and Kvasir, face recognition datasets such as Yale Faces A and Yale Faces B, and object recognition benchmarks such as COIL-20 and COIL-100, all showed that the classification accuracy of CNNs applied to these datasets might be biased [13]. In these studies, a small portion of the background was extracted to create new datasets, which were then utilized to train CNN models. The findings indicate that CNNs achieved high classification accuracy even when trained on datasets lacking relevant information or when systematically influenced by irrelevant features in the image data. The results are displayed in Figure 5.

In summary, CNN prediction accuracy as determined by the standard practices should be treated cautiously, even when high accuracy is achieved, until the input image regions leading to those predictions are visually inspected. It is now evident that many datasets that are used commonly for machine learning experiments can contain some level of bias, and this should be taken into account before relying solely on high-accuracy figures. While the presence of such bias can be tested by using a blank background from each image as carried out in [13,29], in some cases, background parts of images might not be available. This paper proposes a method that involves the application of several image transforms, including Fourier, wavelet transforms, and median filter, and their combination. These transforms were applied to identify bias and recover background noise information used

by CNN to classify images. The method is applied to the whole images, and can therefore be used in cases where no blank background is available to test for CNN bias.

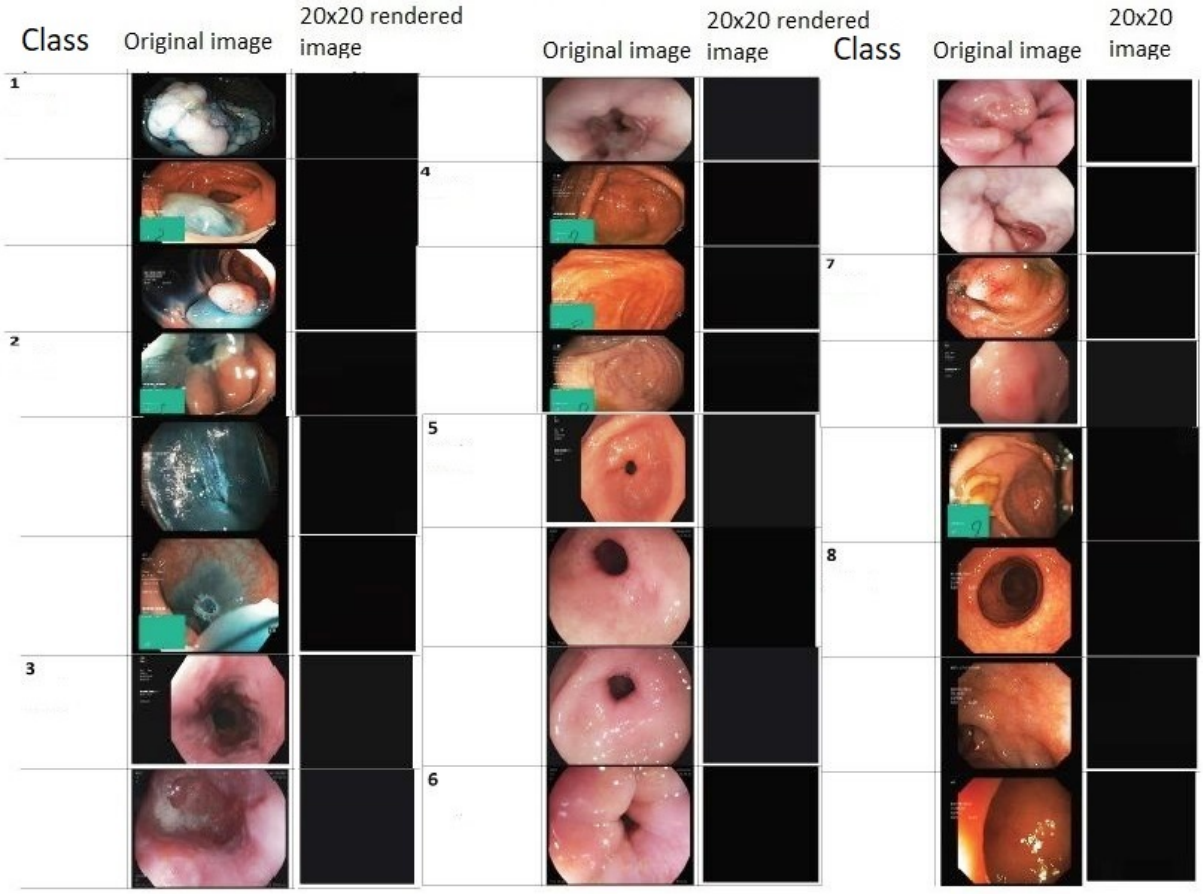


Figure 4. Original images from KVASIR and the 20 × 20 portion of the top left corner separated from the original images [13].

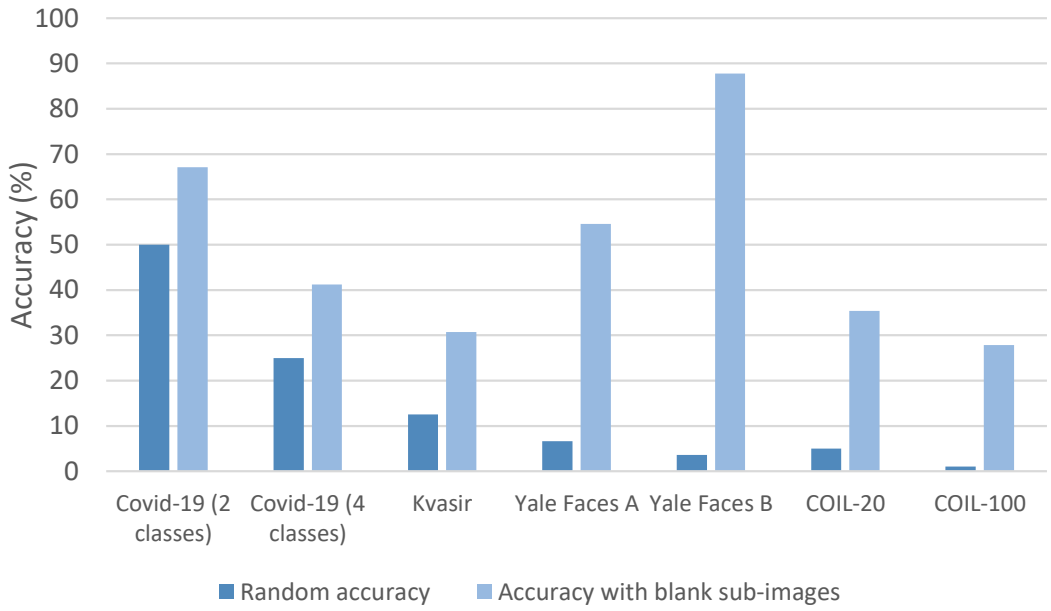


Figure 5. Classification accuracy of CNN models trained and tested on seemingly blank sub-images taken from image background of several common image benchmark datasets [13].

3. Data and CNN Architectures

The experiments described in this paper were carried out across six distinct datasets to ensure that the findings were robust across different contexts. As shown in Figure 6, datasets used in this study can be further categorized into three types: natural datasets, non-natural or synthetic datasets, and mixed or hybrid datasets.

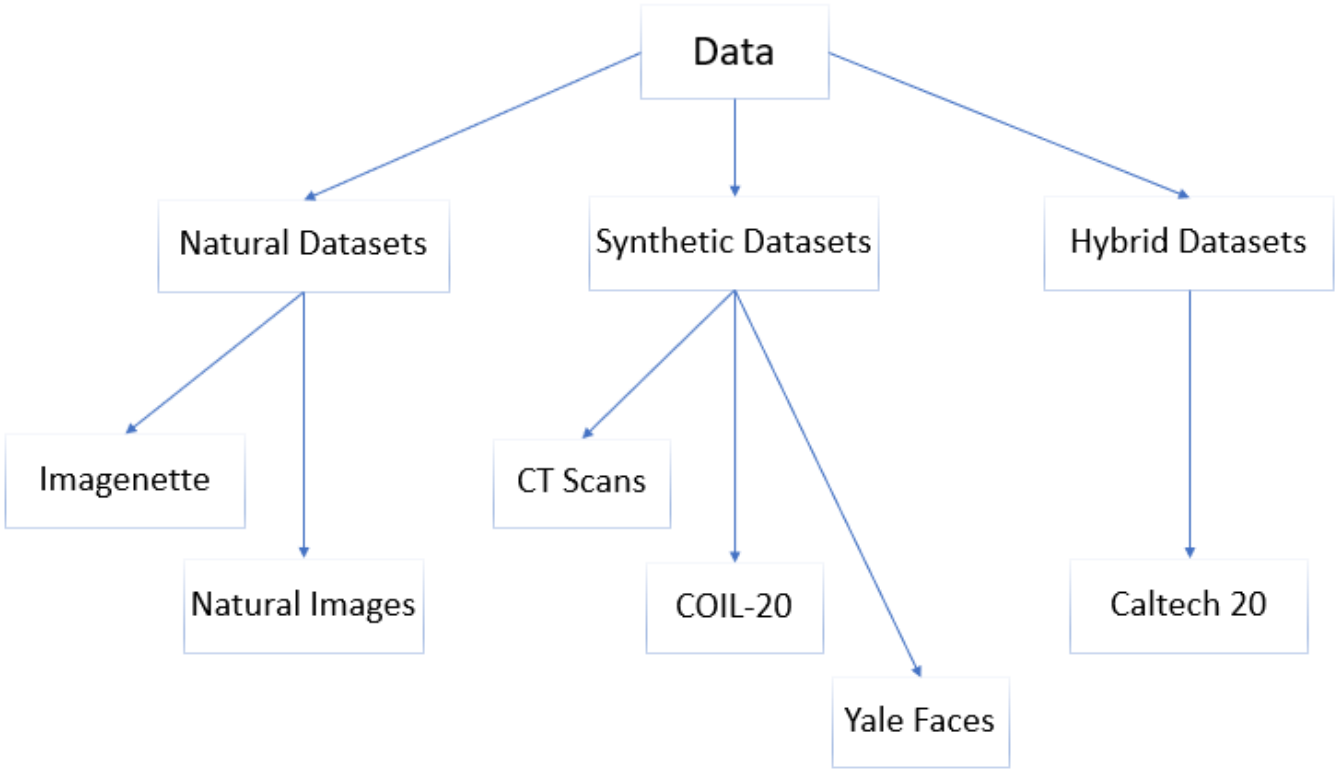


Figure 6. Categorization of data. The data used in this paper include natural images collected from various sources, as well as controlled datasets such that all images in the dataset are provided from a single source.

3.1. Natural Datasets

Natural datasets comprise real-world images collected from a high number of different sources, and exhibit high variability in lighting, angles, and object appearances. Two such naturally collected datasets were employed in this study:

Imagenette: A commonly used smaller subset of the *ImageNet* [32] dataset, containing 10 different classes. The dataset contains images collected from a broad variety of sources, and is very commonly used in machine learning experiments. Imagenette is accessible at https://github.com/fastai/imagenette (accessed on 12 December 2024).

Natural Images: This dataset consists of images coming from eight different classes collected from several sources [34]. The classes include airplane, car, cat, dog, flower, fruit, motorbike, and person. The class "fruit" is discarded in this research because the background was changed to white, which could be used to identify that class and therefore could introduce biases in the classification process.

3.2. Synthetic Datasets

These datasets were either synthetically generated or created under controlled conditions, exhibiting less variability compared to natural datasets. Because of the lower variability and the controlled process, these datasets have higher vulnerability to bias when

analyzed by machine learning [13,28,29]. Therefore, these datasets can be used as datasets that are known as datasets that contain information leading to bias, and any method for bias identification needs to be sensitive to the bias in these datasets. Following are the three different synthetic datasets used:

CT Scans: Contains lung CT scan images across four classes, which are *normal*, *COVID-19*, *bacterial pneumonia*, and *viral pneumonia*. The dataset has been shown to contain certain information that can significantly influence CNN's learning regardless of its medical relevance [30].
COIL-20: The Columbia Object Image Library [35] is a repository of gray-scale pictures that contains such twenty objects. All objects could be precisely put on a turntable. The turntable was turned at 360 degrees until a change of object was appropriately visible from a static camera. Images of the objects were captured in 5-degree angular orientations. Model and Shamir [29] demonstrated that *COIL-20*, being generated in a controlled environment, contains biases that significantly affect the model's classification accuracy. The exceptionally high classification accuracy, even when the object was absent from the test image, means that the model learned patterns from bias or background that are irrelevant to the detection of the objects. The presence of such bias makes this dataset useful for the purpose of this study, aiming to identify bias in image datasets.

Yale Faces: The Yale Face Database [36,37] contains 165 images of 15 subjects, with different facial expressions and lighting conditions. It includes the facial expressions of happy and sad, with and without glasses, sleepy, normal, wink, and surprised. It also includes lighting from different directions. Dhar and Shamir [13] assessed face recognition benchmark datasets, including *Yale Faces*, to evaluate CNN performance. Similarly to COIL-20, the findings reveal a consistent bias within the dataset, with the CNN achieving notably higher classification accuracy when only a part of the image was used, highlighting the significant extent of this bias.

3.3. Mixed or Hybrid Datasets

Hybrid datasets are collections of images that combine both real-world (natural) images and synthetically generated or modified images. The purpose of these datasets is to leverage the advantages of both types of data to improve the performance and generalization of models.

The *Caltech 20* database is a subset of the larger *Caltech 256* dataset [38]. This study used only the first 20 object categories to simplify the dataset. Interestingly, upon closer inspection, we found that the first 20 classes feature a combination of real-world images and either graphically generated or have had their backgrounds altered to a plain white/black backdrop. Therefore, *Caltech 20* is categorized as a mixed dataset in this study.

3.4. CNN Architecture

If a certain CNN architecture can classify images with background information alone, this is an indication that the dataset contains bias. Such bias introduces a risk that any CNN architecture tested on the data can be driven by that bias. This study utilized the very commonly used Visual Geometry Group 16 (VGG16) neural network architecture [39], which has demonstrated excellent classification accuracy. For instance, the architecture achieved high classification accuracy of ∼92.77% on the full ImageNet dataset [39]. VGG16 comprises 13 convolutional layers, 5 max-pooling layers, and 3 fully connected (dense) layers, adding up to 21 layers. Only 16 of these layers have weights. The network takes as input images of size 224 × 224 with three RGB channels. Figure 7 visualizes the VGG15 architecture. In all cases, 70% of the samples were allocated for the training set, 15% for the test set, and 15% for the validation set. The training was with 40 epochs and Adam optimizer. The code used for the experiments can be found at https://github.com/SaiTeja-Erukude/identifying-bias-in-dnn-classification (accessed on 12 December 2024).

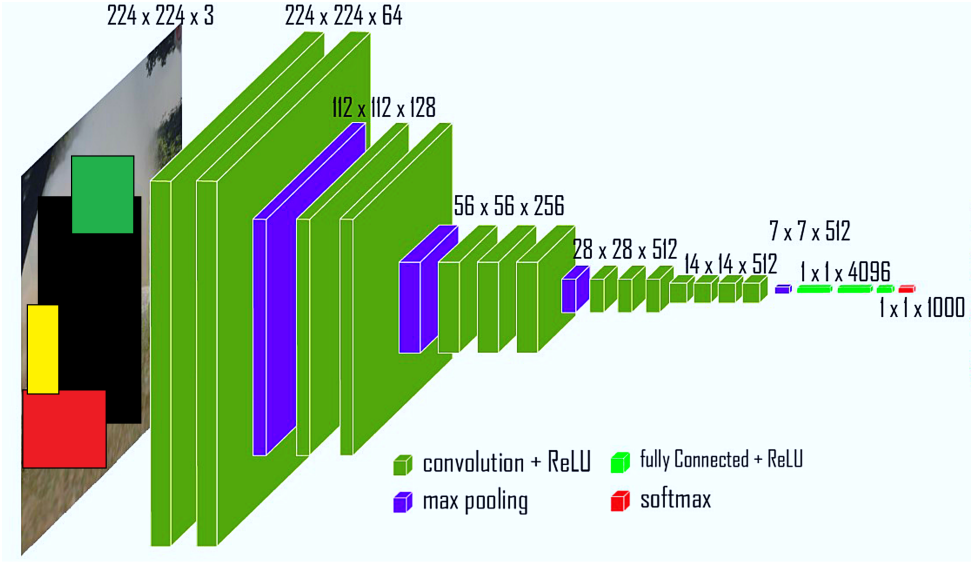


Figure 7. VGG16 architecture.

4. Methods and Results

As demonstrated in Section 2, CNNs are capable of identifying the correct class even when they are trained and tested with seemingly blank background parts of the images. The presence of such bias can be identified by performing experiments similar to the experiments shown in Section 2, repeating the same machine learning process using seemingly blank parts taken from the background of each image. But this approach can be used only if the images in the dataset have such a blank background. If the entire image contains information that is meaningful for the classification, a control experiment with blank background cannot be performed, since better-than-random classification accuracy can be attributed to the visually relevant image content rather than the bias. Here, we approach the task by applying image transforms to the original images, and testing the classification accuracy with the transformed images to identify the presence of biases.

4.1. Fourier Transform

Fourier transform is one of the most widely employed methods in image processing, used to break down an image into its sine and cosine frequency components [40]. This transform converts the input spatial domain image to its equivalent frequency domain image. In the resulting image, each pixel denotes a certain frequency present in the spatial domain image. Since Discrete Fourier Transform (DFT) is a sampled version, it does not contain all frequencies forming an image. It is based only on a set of samples good enough to fully describe the image in the spatial domain. The number of frequencies equals the number of pixels in the spatial domain image—i.e., the image in the spatial and Fourier domains are of the same size [41].

VGG16 models were trained and tested on the dataset after applying the Fourier transform to all images. Figures 8 and 9 show the classification accuracy of the full images after applying the Fourier transform, and the classification accuracy when using the 20×20 sub-images cropped from the top right corner of each original image and applying the Fourier transform. As the figures show, applying the Fourier transform reduced the classification accuracy of the CNN across all the datasets, both for the full images and for the cropped sub-images.

For example, in the Imagenette dataset, the accuracy dropped significantly from ∼59% with the original full images to ∼38% for Fourier-transformed full images. The accuracy for cropped sub-images dropped to around random levels. Similarly, the COIL-20 dataset saw its accuracy decline from ∼100% to ∼80%. As the figures show, most other datasets also showed a decrease in classification accuracy for both full and cropped images.

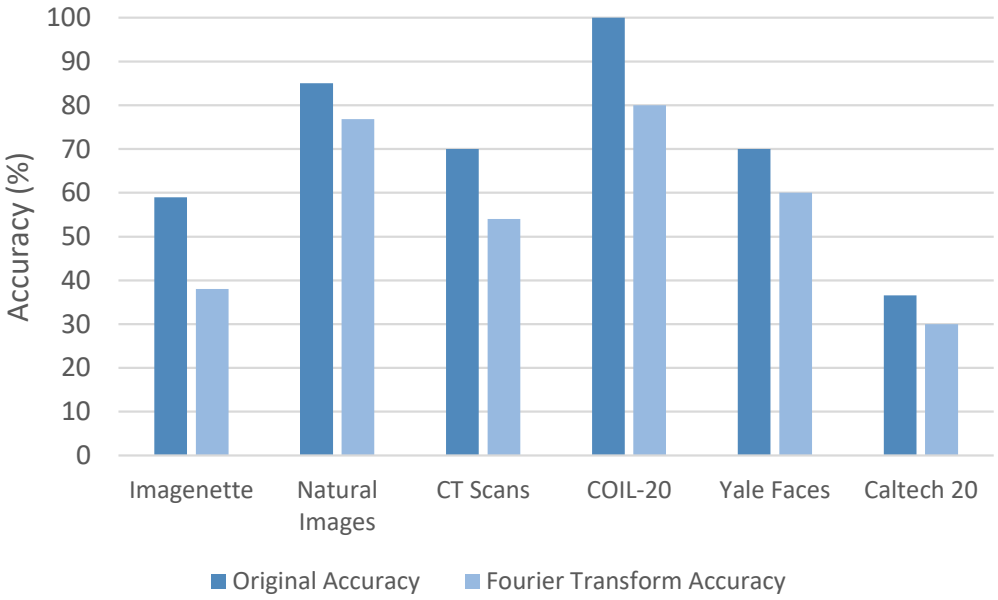


Figure 8. Classification accuracy when using the Fourier-transformed full images.

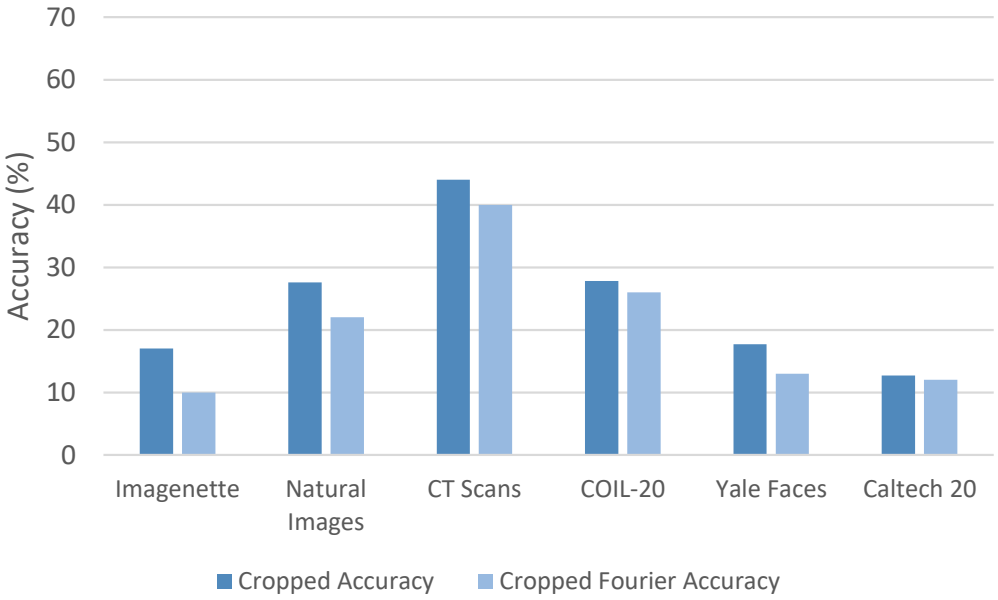


Figure 9. Classification accuracy when using the Fourier-transformed small sub-images taken from the background.

One possible explanation for this decrease could be that transforming images into the Fourier domain leads to a loss of spatial information as sensed by the VGG16 architecture. This is because VGG16 is not designed for the frequency domain, especially given that it is based on convolutional layers that are sensitive to pixels that are close to each other in the image. CNNs rely on this spatial information to effectively learn and classify features. Yet, using the Fourier-transformed images makes it challenging to determine whether the drop in classification accuracy is due to the loss of relevant features or driven by the reduction of inherent bias in the datasets that were influencing the CNN's predictions. Therefore, the Fourier transform is an example of a transform that does not effectively help distinguish between biases in natural and non-natural datasets, as it reduces accuracy across all types of datasets.

4.2. Wavelet Transform

Wavelet transforms are mathematical methods used for investigating and retrieving data from images [42,43]. They break an image into components at various positions, preserving both frequency and spatial data. In contrast to the Fourier transform, which outputs a global frequency, the wavelet transform provides a time–frequency (or space–frequency) representation, allowing for localized evaluation in both space and frequency domains.

Wavelet transform was applied to all datasets. As was carried out with the Fourier transform, the experiments included the classification accuracy using the full images, as well as experiments with the 20 × 20 cropped sub-images. The wavelet transformations were generated by using the *PyWavelets* Python package [44]. Two commonly used wavelets were used in this study: Haar and Daubechies. Haar wavelets are discontinuous and seem like a step function. They are a simple form of wavelet that can be used to test the impact of wavelets in the broad sense. Daubechies wavelets best represent polynomial trends, and are used often in situations of signal discontinuity, which is a situation that can be used to separate between foreground signal and detectable background "noise".

In this study, the wavelets were implemented through the two dimensional wavelet transform function (dwt2) of the *Wavelet Transforms in Python* library (PyWavelets) with the "haar" parameter and the "daubechies" parameter. The code used for the transform is available at https://github.com/SaiTeja-Erukude/identifying-bias-in-dnn-classification, accessed on 12 December 2024. Figures 10 and 11 show the images produced by applying these two wavelets. The transforms appear comparable to each other using the unaided human eye.

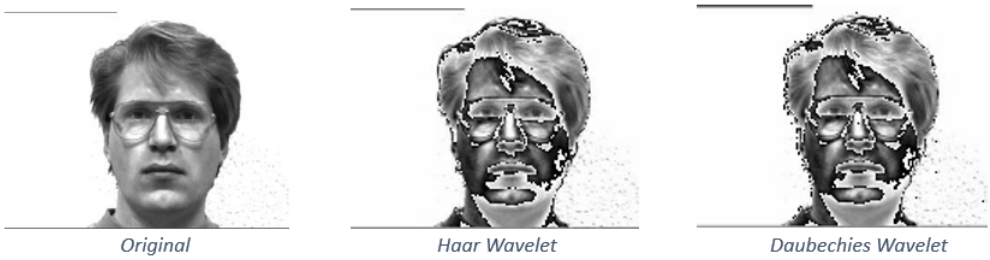


Figure 10. Example of an original image and the Haar and Daubechies Discrete Wavelet transformed images.



Figure 11. Example of an original image taken from ImageNette, and the Haar and Daubechies Discrete Wavelet transformed images.

The wavelet findings indicate a notable decline in classification accuracy for natural datasets, while non-natural datasets see either consistent or improved accuracy rates. This trend is observed in both full images and cropped images. For instance, as shown in Figure 12, the prediction accuracy for the Imagenette dataset has dropped from ∼59% to around ∼50% for full images, and from ∼17% to roughly ∼13% (nearly random levels) for cropped images. In contrast, for the COIL-20 dataset, which is synthetic, the accuracy maintains a constant level of 100%, and its cropped image accuracy increases from ∼27.8% to about ∼31.4%.

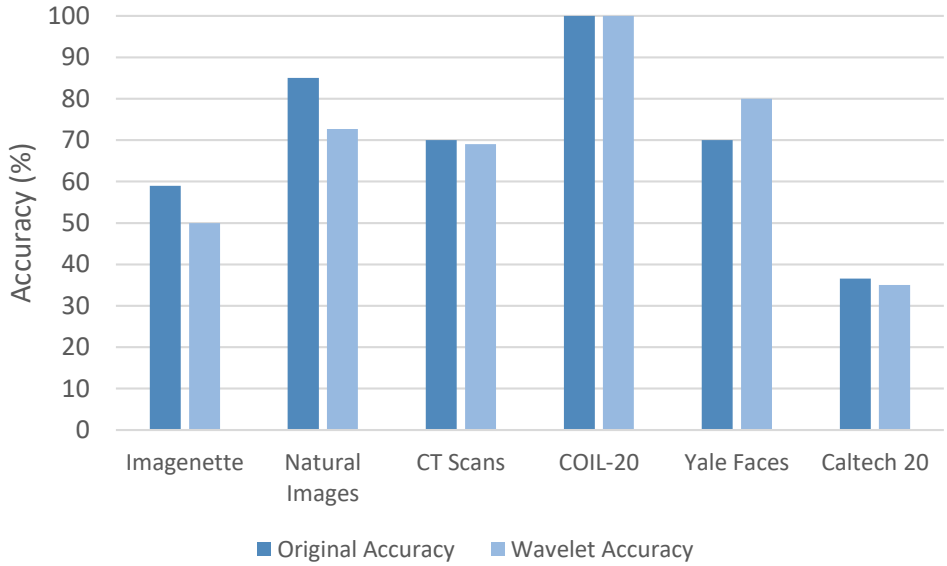


Figure 12. Classification accuracy when using images after applying the wavelet transform.

The 20 × 20 cropped images are seemingly blank and do not contain any visual information that can be sensed by the human eye. However, they still contain hidden information that can influence a CNN model's predictions. As shown in Figure 13, applying a wavelet transform to these cropped images increased their accuracy compared to the accuracy using the original images. This indicates that the wavelet transform revealed and enhanced the hidden signal in the background. That means that CNN has learned to use the background information, leading to more biased and therefore less reliable results.

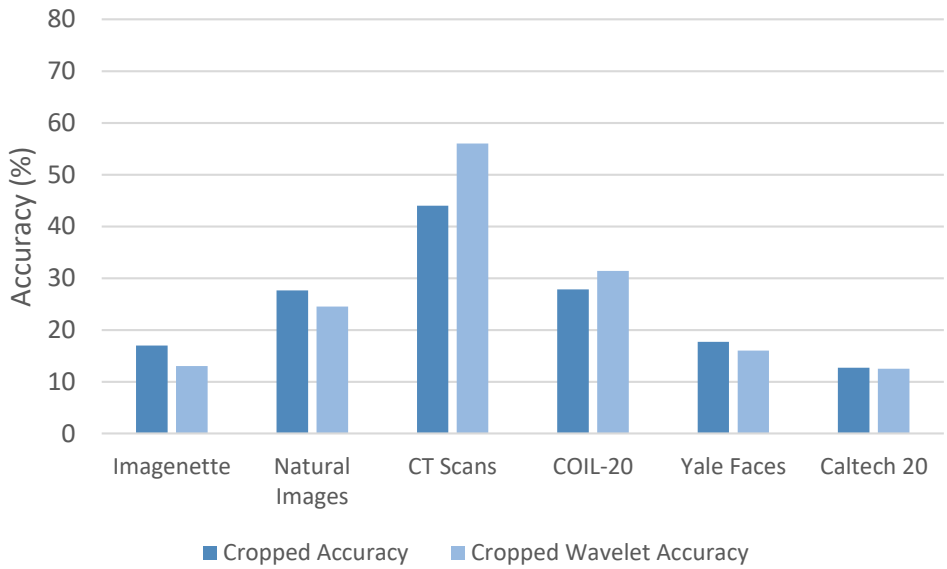


Figure 13. Classification accuracy when using the wavelet transformed cropped images.

In contrast, natural datasets made of images collected from various sources typically do not contain bias in the imaging process, as images of the same class are rarely taken during the same session or provided by the same single source. Therefore, these datasets are affected differently by the wavelet transform. As a result, the classification accuracy for the Imagenette and Natural image datasets do not improve when the wavelet transform is applied. Therefore, the wavelet transform effectively differentiates between contextual information and the presence of background information that leads to bias. It can therefore

be used to test for the possible presence of background information that can lead to bias, yet without confusing it with contextual information.

4.3. Median Filter

The median filter is a very common non-linear simple image processing method that evaluates the image pixel-by-pixel and replaces each pixel with the median of its adjacent entries. The pattern of nearby pixels depicts a window that slides across the entire image, one entry at a time [45]. It is a smoothing technique that effectively removes disturbances from a noisy image or a signal. Unlike low-pass finite impulse response (FIR) filters, median filters are notable for retaining image edges, making them commonly used in image processing. Median filtering was performed on all datasets described in Section 3 using OpenCV's "medianBlur()" [46] method with a window size of 5×5 to smooth out the images and reduce noise.

For full images, the classification accuracy decreased when using natural datasets with the median filter. For example, Imagenette dropped from ∼59% to about ∼55%, and Natural Images fell from ∼85% to about ∼81.6%. In contrast, the accuracy for synthetic datasets has remained stable, while hybrid datasets improved. For instance, the accuracy for Caltech 20 increased from around ∼36.6% to around ∼39%.

The median filter had a relatively small window of 5×5. The size of the window did not significantly affect the classification accuracy. Only when the window was larger than 25×25 did the classification accuracy start to decrease, but for smaller window sizes, the results were nearly identical. For instance, a window size of 7×7 or 9×9 can be used with no substantial impact on the results.

The median filters were also tested in combination with the wavelet transform, such that the wavelet transform was applied after applying the median filter. As before, the analysis was applied to the full images, as well as the cropped images. The accuracy for natural datasets, such as Imagenette and Natural Images, decreased by about ∼9.5% and ∼14.8%, respectively. The classification accuracy for synthetic and hybrid datasets either remained stable or increased. This indicates that when applying the median filter followed by a wavelet transform, the classification accuracy when using the datasets that are known to contain bias increases or remains the same, while the accuracy of datasets collected from various sources drops. Figures 14 and 15 show the classification accuracy when applying the median filter to the full images and to the cropped sub-images, respectively. Similarly, Figures 16 and 17 show the classification accuracy when applying the median filter and wavelet transform to the original full images and to the cropped sub-images.

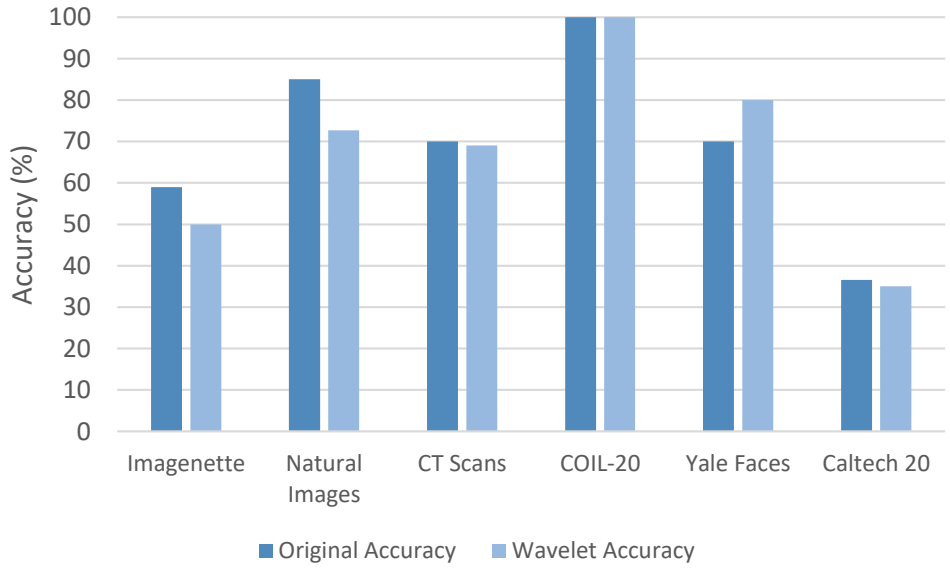


Figure 14. Classification accuracy when using full images after applying median filtering.

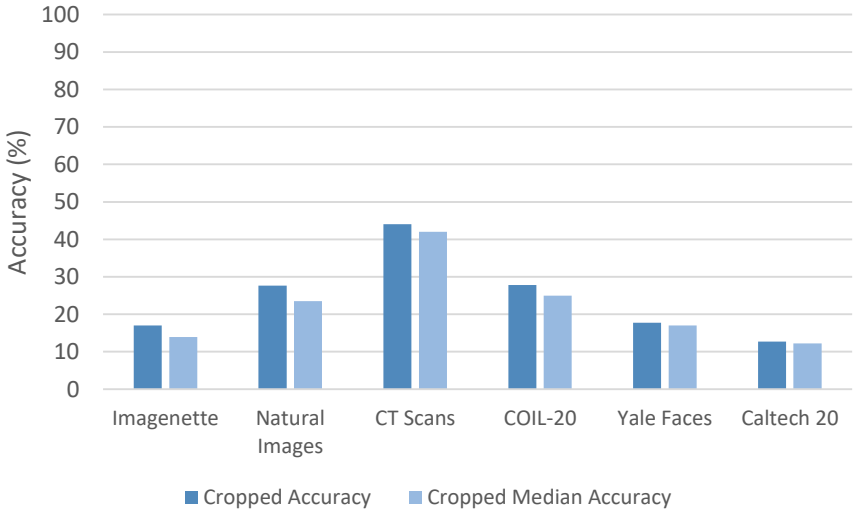


Figure 15. Classification accuracy when using sub-images of blank background after applying median filter.

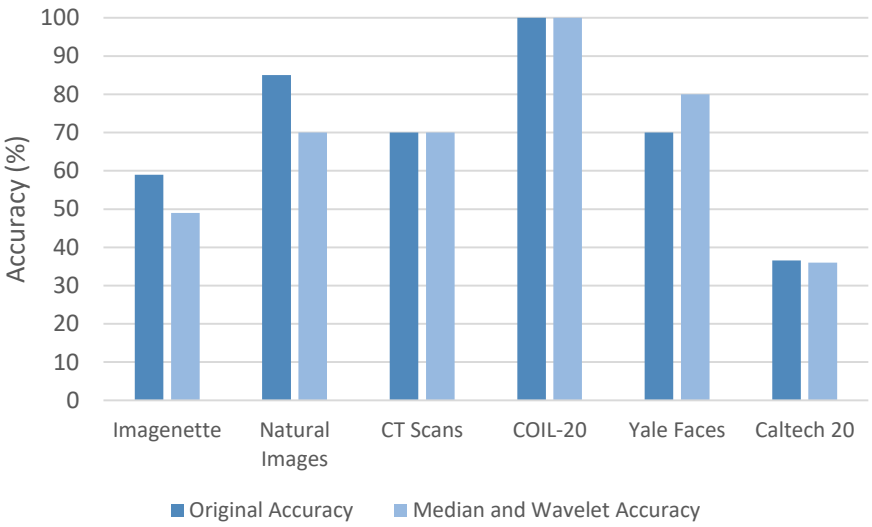


Figure 16. Classification accuracy of full images after applying both median and wavelet transforms.

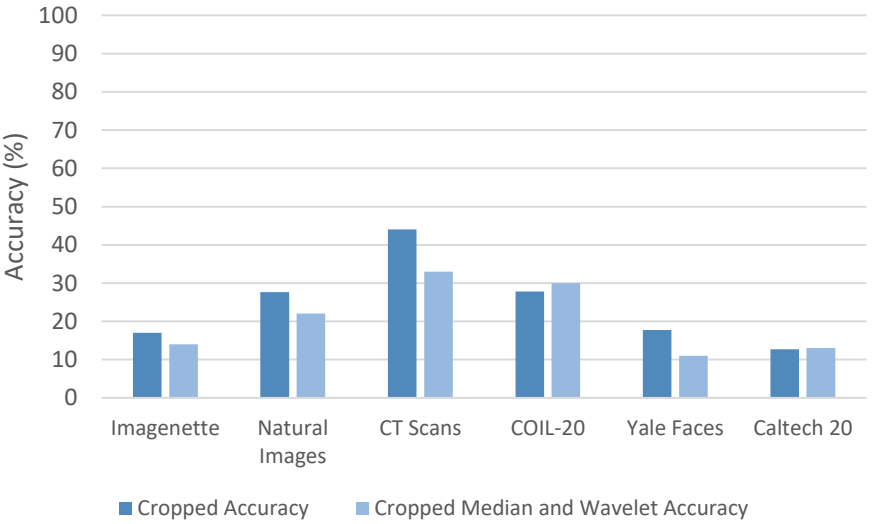


Figure 17. Classification accuracy of blank background sub-images after applying both median and wavelet transforms.

4.4. Summary of the Results

Table 1 summarizes the results discussed in Section 4. The table shows the classification accuracy, as well as the specificity, sensitivity, and F1 score. The table provides a higher resolution view of the results.

Table 1. The classification accuracy, F1 scores, precision, and recall of the experiments.

Dataset	Original Images	Fourier Transformed	Wavelet Transformed	Median Filtered	Median + Wavelet
Imagenette	Accuracy: 59% Precision: 57.5% Recall: 59.6% F1: 0.5853	Accuracy: 38% Precision: 37.1% Recall: 38.2% F1 Score: 0.3764	Accuracy: 50% Precision: 48.3% Recall: 49.7% F1: 0.4899	Accuracy: 55% Precision: 52.3% Recall: 53.4% F1: 0.5284	Accuracy: 49.5% Precision: 47% Recall: 50% F1: 0.4845
Natural Images	Accuracy: 85% Precision: 86% Recall: 85.33% F1: 0.8552	Accuracy: 76.8% Precision: 77.3% Recall: 76.6% F1 Score: 0.7660	Accuracy: 72.8% Precision: 73.7% Recall: 73.5% F1 Score: 0.7261	Accuracy: 81.6% Precision: 82% Recall: 81.3% F1 Score: 0.8157	Accuracy: 70.6% Precision: 72.7% Recall: 70.7% F1 Score: 0.7134
CT Scans	Accuracy: 70% Precision: 66.7% Recall: 70.6% F1: 0.6859	Accuracy: 54% Precision: 52.1% Recall: 54.4% F1 Score: 0.5323	Accuracy: 69% Precision: 67.2% Recall: 69.1% F1: 0.6814	Accuracy: 69% Precision: 67% Recall: 68.8% F1: 0.6789	Accuracy: 70.5% Precision: 67.3% Recall: 70.9% F1: 0.6905
COIL-20	Accuracy: 100% Precision: 100% Recall: 100% F1: 1	Accuracy: 80% Precision: 81.2% Recall: 80% F1 Score: 0.8060	Accuracy: 100% Precision: 100% Recall: 100% F1: 1	Accuracy: 100% Precision: 100% Recall: 100% F1: 1	Accuracy: 100% Precision: 100% Recall: 100% F1: 1
Yale Faces	Accuracy: 70% Precision: 75.4% Recall: 70% F1: 0.7260	Accuracy: 60% Precision: 63.8% Recall: 60.4% F1 Score: 0.6205	Accuracy: 80% Precision: 82.2% Recall: 80.1% F1: 0.8114	Accuracy: 70% Precision: 74.1% Recall: 71.3% F1: 0.7267	Accuracy: 80% Precision: 83% Recall: 80.7% F1: 0.8183
Caltech 20	Accuracy: 36.6% Precision: 30.4% Recall: 35.2% F1: 0.3262	Accuracy: 30% Precision: 25.3% Recall: 29.1% F1: 0.2707	Accuracy: 35% Precision: 33.3% Recall: 34.8% F1: 0.3403	Accuracy: 39% Precision: 37.1% Recall: 38.6% F1: 0.3784	Accuracy: 36% Precision: 32% Recall: 35% F1: 0.3343

5. Conclusions

CNNs have become the most prevalent computational tool for image classification. Due to their black-box nature, reliable performance evaluation can become challenging, and biases or unreliable performance evaluations can go unnoticed. In this study, several techniques including Fourier transforms, wavelet transforms, simple median filtering, and their combinations were tested. The wavelet transform, simple median filtering transform, and their combination proved to be effective in distinguishing between biases in natural datasets that are driven by contextual information, and bias in non-natural datasets that has been shown in previous studies to be driven by artifacts acquired through the imaging process. Therefore, the results show that the method can distinguish between contextual information and imaging process bias, and lay bare the presence of such bias even without the need to separate background information from the original images.

Applying a median filter smoothens the image out and eliminates noise, and subsequently applying wavelet transform to this smoothed dataset further enhances the signal in the image. This process affects the contextual visual information in a different manner than it affects the artifacts acquired through the imaging process. The wavelet transform highlights any hidden signals introduced during the image acquisition process. In summary, these methods decreased the prediction accuracy for natural datasets but improved the accuracy for non-natural datasets, effectively addressing the distinct biases present in each.

Such methods should be applied to verify the soundness of CNNs when they are used to classify image datasets, and specifically datasets acquired through a controlled environment

such medical images. Once the neural network is trained, it is tested by classifying the images after pre-processing them as described in Section 4. This can provide an indication of whether bias is present. This is needed only in cases where no parts of the background that are clearly irrelevant to the image classification task are available, as the classification of such 20×20 blank parts of the images can provide a simpler indication of the presence of bias.

Testing for bias using the proposed method is straightforward: First, the images are transformed using the image transformations shown in Section 4. The transformed images are generated from the original full-size images, and not the 20×20 background sub-images; as such, background sub-images may not be available for every dataset. After transforming the images, the CNN is applied to train and test the classification accuracy with the transformed images. If the classification accuracy with the transformed images is similar or higher than the classification accuracy when using the original images, it is an indication that the classification might be driven by bias.

It must be reminded that the method can identify the presence of bias that is difficult to notice, but biases can have different forms and reasons. Therefore, if the proposed method does not identify the presence of bias, this does not necessarily ensure that no bias exists. Other biases can also be present, some of them might be unknown or unexpected, and the proposed method is not designed to identify all possible biases.

The proposed method also does not correct for the bias, but merely identifies its existence. As demonstrated in Section 2, numerous experiments were conducted without noticing the relevant bias, and many commonly used benchmark datasets can be classified with high classification accuracy by just using a seemingly blank sub-image of the background. Therefore, before fully trusting the prediction accuracy of CNNs, it is recommended to apply methods such as the tests outlined in Section 4 to identify possible biases in the datasets, which should also help in distinguishing between contextual information and the presence of background bias. This evaluation helps gauge CNNs' reliability and ensures their accuracy and trustworthiness.

Future work will focus on identifying the sources of biases, and potentially altering images to correct for biases. Correction of biases can make use of GANs or adversarial neural networks to create images that visually seem identical to the original images, but are handled differently by deep neural networks. Such pre-processing can not only identify bias, but also help in avoiding it by removing information that is irrelevant to the problem at hand but can still be used by classifiers to classify the images.

Author Contributions: Conceptualization, S.T.E., A.J. and L.S.; methodology, S.T.E. and L.S.; software, S.T.E.; validation, S.T.E.; formal analysis, S.T.E. and L.S.; investigation, S.T.E., A.J. and L.S.; resources, S.T.E., A.J. and L.S.; data curation, S.T.E.; writing—original draft preparation, S.T.E. and L.S.; writing—review and editing, S.T.E., A.J. and L.S.; visualization, S.T.E.; supervision, L.S.; project administration, L.S.; funding acquisition, L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported in part by NSF grant 2148878.

Data Availability Statement: The complete Python code utilized in this research study is available in a GitHub repository: https://github.com/SaiTeja-Erukude/identifying-bias-in-dnn-classification, accessed on 12 December 2024. Using the code will require changing the input and output directories. Required libraries are pywt, cv2, and keras.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Uchida, S.; Ide, S.; Iwana, B.K.; Zhu, A. A further step to perfect accuracy by training CNN with larger data. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 405–410.
2. Santos, C.F.G.D.; Papa, J.P. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Comput. Surv.* **2022**, *54*, 1–25. [CrossRef]

3. Thanapol, P.; Lavangnananda, K.; Bouvry, P.; Pinel, F.; Leprévost, F. Reducing overfitting and improving generalization in training convolutional neural network (CNN) under limited sample sizes in image recognition. In Proceedings of the International Conference on Information Technology, Xi'an, China, 25–27 December 2020; pp. 300–305.
4. Gavrilov, A.D.; Jordache, A.; Vasdani, M.; Deng, J. Preventing model overfitting and underfitting in convolutional neural networks. *Int. J. Softw. Sci. Comput. Intell.* **2018**, *10*, 19–28. [CrossRef]
5. Sikha, O.; Benitez, R. Deep Feature Discriminability as a Diagnostic Measure of Overfitting in CNN Models. In Proceedings of the Science and Information Conference, London, UK, 26–27 June 2024; pp. 218–233.
6. Pastaltzidis, I.; Dimitriou, N.; Quezada-Tavarez, K.; Aidinlis, S.; Marquenie, T.; Gurzawska, A.; Tzovaras, D. Data augmentation for fairness-aware machine learning: Preventing algorithmic bias in law enforcement systems. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, 21–24 June 2022; pp. 2302–2314.
7. Raja Babu, K.A.; Sathish, R.; Pattanaik, M.; Venkataramani, R. Synthetic Simplicity: Unveiling Bias in Medical Data Augmentation. In Proceedings of the MICCAI Workshop on Data Engineering in Medical Imaging, Marrakesh, Morocco, 11 October 2024; pp. 64–72.
8. Richter, M.L.; Byttner, W.; Krumnack, U.; Wiedenroth, A.; Schallner, L.; Shenk, J. (Input) size matters for CNN classifiers. In Proceedings of the 30th International Conference on Artificial Neural Networks ICANN 2021, Bratislava, Slovakia, 14–17 September 2021; Part II 30; pp. 133–144.
9. Ball, P. Is AI leading to a reproducibility crisis in science? *Nature* **2023**, *624*, 22–25. [CrossRef] [PubMed]
10. Wang, B.; Ma, R.; Kuang, J.; Zhang, Y. How Decisions Are Made in Brains: Unpack "Black Box" of CNN with Ms. Pac-Man Video Game. *IEEE Access* **2020**, *8*, 142446–142458. [CrossRef]
11. Buhrmester, V.; Münch, D.; Arens, M. Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 966–989. [CrossRef]
12. Tommasi, T.; Patricia, N.; Caputo, B.; Tuytelaars, T. A Deeper Look at Dataset Bias. *arXiv* **2015**, arXiv:1505.01257.
13. Dhar, S.; Shamir, L. Evaluation of the benchmark datasets for testing the efficacy of deep convolutional neural networks. *Vis. Inform.* **2021**, *5*, 92–101. [CrossRef]
14. Alqaraawi, A.; Schuessler, M.; Weiß, P.; Costanza, E.; Berthouze, N. Evaluating saliency map explanations for convolutional neural networks: A user study. In Proceedings of the 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, 17–20 March 2020; pp. 275–285.
15. Kim, B.; Seo, J.; Jeon, S.; Koo, J.; Choe, J.; Jeon, T. Why are saliency maps noisy? Cause of and solution to noisy saliency maps. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop, Seoul, Republic of Korea, 27–28 October 2019; pp. 4149–4157.
16. Arun, N.; Gaw, N.; Singh, P.; Chang, K.; Aggarwal, M.; Chen, B.; Hoebel, K.; Gupta, S.; Patel, J.; Gidwani, M.; et al. Assessing the trustworthiness of saliency maps for localizing abnormalities in medical imaging. *Radiol. Artif. Intell.* **2021**, *3*, e200267. [CrossRef] [PubMed]
17. Simonyan, K. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv* **2013**, arXiv:1312.6034.
18. Pfau, J.; Young, A.T.; Wei, M.L.; Keiser, M.J. Global saliency: Aggregating saliency maps to assess dataset artefact bias. *arXiv* **2019**, arXiv:1910.07604.
19. Hashimoto, N.; Fukushima, D.; Koga, R.; Takagi, Y.; Ko, K.; Kohno, K.; Nakaguro, M.; Nakamura, S.; Hontani, H.; Takeuchi, I. Multi-scale domain-adversarial multiple-instance CNN for cancer subtype classification with unannotated histopathological images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 3852–3861.
20. Lin, Y.; Zhao, H.; Ma, X.; Tu, Y.; Wang, M. Adversarial attacks in modulation recognition with convolutional neural networks. *IEEE Trans. Reliab.* **2020**, *70*, 389–401. [CrossRef]
21. Wang, J.; Zhao, J.; Yin, Q.; Luo, X.; Zheng, Y.; Shi, Y.Q.; Jha, S.K. SmsNet: A new deep convolutional neural network model for adversarial example detection. *IEEE Trans. Multimed.* **2021**, *24*, 230–244. [CrossRef]
22. Pertigkiozoglou, S.; Maragos, P. Detecting adversarial examples in convolutional neural networks. *arXiv* **2018**, arXiv:1812.03303.
23. Jaipuria, N.; Zhang, X.; Bhasin, R.; Arafa, M.; Chakravarty, P.; Shrivastava, S.; Manglani, S.; Murali, V.N. Deflating dataset bias using synthetic data augmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 772–773.
24. Sharma, S.; Zhang, Y.; Ríos Aliaga, J.M.; Bouneffouf, D.; Muthusamy, V.; Varshney, K.R. Data augmentation for discrimination prevention and bias disambiguation. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, 7–8 February 2020; pp. 358–364.
25. McLaughlin, N.; Del Rincon, J.M.; Miller, P. Data-augmentation for reducing dataset bias in person re-identification. In Proceedings of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance, Karlsruhe, Germany, 25–28 August 2015; pp. 1–6.
26. Iosifidis, V.; Ntoutsi, E. Dealing with bias via data augmentation in supervised learning scenarios. *Jo Bates Paul D. Clough Robert Jäschke* **2018**, *24*, 24–29 .
27. Torralba, A.; Efros, A.A. Unbiased look at dataset bias. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 1521–1528.

28. Shamir, L. Evaluation of face datasets as tools for assessing the performance of face recognition methods. *Int. J. Comput. Vis.* **2008**, *79*, 225–230. [CrossRef]
29. Model, I.; Shamir, L. Comparison of Data Set Bias in Object Recognition Benchmarks. *IEEE Access* **2015**, *3*, 1953–1962. [CrossRef]
30. Majeed, T.; Rashid, R.; Ali, D.; Asaad, A. Issues associated with deploying CNN transfer learning to detect COVID-19 from chest X-rays. *Phys. Eng. Sci. Med.* **2020**, *43*, 1289–1303. [CrossRef] [PubMed]
31. Dhar, S.; Shamir, L. Systematic biases when using deep neural networks for annotating large catalogs of astronomical images. *Astron. Comput.* **2022**, *38*, 100545. [CrossRef]
32. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
33. Winastwan, R. Class Activation Mapping (CAM): Better Interpretability in Deep Learning Models. Natural Language Processing (NLP) Advanced Guide. 25 March 2024. Available online: https://zilliz.com/learn/class-activation-mapping-CAM (accessed on 12 December 2024).
34. Roy, P.; Ghosh, S.; Bhattacharya, S.; Pal, U. Effects of Degradations on Deep Neural Network Architectures. *arXiv* **2018**, arXiv:1807.10108.
35. Nene, S.A.; Nayar, S.K.; Murase, H. *Columbia Object Image Library (COIL-20)*; Technical Report No. CUCS-006-96, Citeseer; Columbia University: New York, NY, USA, 1996.
36. Georghiades, A.S.; Belhumeur, P.N.; Kriegman, D.J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 643–660. [CrossRef]
37. Belitskaya, O. Yale Face Database—Kaggle. 2018. Available online: https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database (accessed on 19 November 2024)
38. Griffin, G.; Holub, A.; Perona, P. *Caltech-256 Object Category Dataset*; Technical Report 7694; California Institute of Technology Pasadena: Pasadena, CA, USA, 2007.
39. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv: 1409.1556.
40. Cochran, W.T.; Cooley, J.W.; Favin, D.L.; Helms, H.D.; Kaenel, R.A.; Lang, W.W.; Maling, G.C.; Nelson, D.E.; Rader, C.M.; Welch, P.D. What is the fast Fourier transform? *Proc. IEEE* **1967**, *55*, 1664–1674. [CrossRef]
41. Fisher, R.; Perkins, S.; Walker, A.; Wolfart, E. *Hypermedia Image Processing Reference*; John Wiley & Sons Ltd: London, UK, 1996; pp. 209–213.
42. Agarwal, S.; Singh, O.; Nagaria, D. Analysis and Comparison of Wavelet Transforms for Denoising MRI Image. *Biomed. Pharmacol. J.* **2017**, *10*, 831–836. [CrossRef]
43. Othman, G.; Zeebaree, D.Q. The applications of discrete wavelet transform in image processing: A review. *J. Soft Comput. Data Min.* **2020**, *1*, 31–43.
44. PyWavelet Developers. PyWavelets Documentation. 2024. Available online: https://pywavelets.readthedocs.io/en/latest/ (accessed on 12 December 2024).
45. Dhanani, S.; Parker, M. Introduction to Digital Filtering. In *Digital Video Processing for Engineers*; Dhanani, S., Parker, M., Eds.; Newnes: Oxford, UK, 2013; pp. 19–28. [CrossRef]
46. OpenCV. OpenCV Image Filtering Documentation. 2024. Available online: https://docs.opencv.org/4.x/d4/d86/group__imgproc__filter.html (accessed on 19 November 2024)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.