IFBiD: Inference-Free Bias Detection

Ignacio Serna, Daniel DeAlcala, Aythami Morales, Julian Fierrez, Javier Ortega-Garcia

Biometrics and Data Pattern Analytics Lab (BiDA-Lab), Autonomous University of Madrid
{ignacio.serna, daniel.dealcala, aythami.morales, julian.fierrez, javier.ortega}@uam.es

Abstract

This paper is the first to explore an automatic way to detect bias in deep convolutional neural networks by simply looking at their weights, without the model inference for a specific input. Furthermore, it is also a step towards understanding neural networks and how they work. We analyze how bias is encoded in the weights of deep networks through a toy example using the Colored MNIST database and we also provide a realistic case study in gender detection from face images using state-of-the-art methods and experimental resources. To do so, we generated two databases with 36K and 48K biased models each. In the MNIST models we were able to detect whether they presented strong or low bias with more than 99% accuracy, and we were also able to classify between four levels of bias with more than 70% accuracy. For the face models, we achieved 83% accuracy in distinguishing between models biased towards Asian, Black, or Caucasian ethnicity.

Introduction

Artificial intelligence is generating more and more expectations. But is it really living up to those expectations? Its use is being reviewed in all areas, from natural language processing for virtual assistants, to computer vision for citizen monitoring systems or medical follow-up (Stone et al. 2016). Deep Neural Networks play a key role in the deployment of machine learning models in these applications. But although these algorithms achieve impressive prediction accuracies, their structure makes them very opaque. Data-driven learning processes make it difficult to control the factors and understand the information from the input data that actually drive their decisions. In this environment new efforts are being devoted to making systems more understandable and interpretable by humans (Mahendran and Vedaldi 2015; Montavon, Samek, and Müller 2018; Bau et al. 2020). More concretely, new techniques are being developed to understand and visualize what machine learning models learn (Zeiler and Fergus 2014; Koh and Liang 2017), as well as models that generate text-based explanations of the decisions they make (Barredo Arrieta et al. 2020; Ortega et al. 2021).

On the other hand, thanks to adequate public outreach and debate, more and more investigations are emerging that uncover some erratic and biased behaviors of these artificial intelligence systems. These errors and biases are calling into question the safety of AI systems, both because of privacy issues (Fierrez, Morales, and Ortega-Garcia 2021) and unintended side effects (Serna et al. 2020).

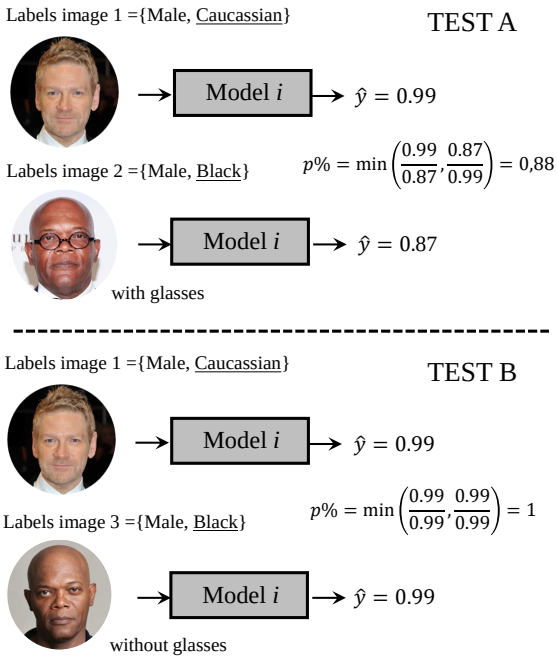

Figure 1: Traditional bias detection test based on inference analysis and the demographic parity measure $p\%$ (Zhang, Lemoine, and Mitchell 2018). The Model i is a gender classifier and the sensitive attribute z is ethnicity. In this example, Test A suggests that the Model i is biased with respect to ethnicity. Test B revealed that the difference in $p\%$ is caused not by ethnicity but by glasses.

One way to build trust into AI systems is to relate their inner workings to human-interpretable concepts (Bau et al. 2020). But research is showing that not all representations in the convolutional layers of a DNN correspond to natural parts, raising the possibility of a different decomposition of the world than humans might expect, requiring further study

into the exact nature of the learned representations (Yosinski et al. 2015; Geirhos et al. 2019).

In this regard, bias detection is a major challenge to ensure trust in machine learning and its applications (Ntoutsi et al. 2020; Terhorst et al. 2022). Recent approaches for bias detection focus on the analysis of model outcomes or the visualization of learned features at the data input level (Alvi, Zisserman, and Nellåker 2018; Zhang, Wang, and Zhu 2018). That is, they are data-bound and need inference to gain insight (see Fig. 1). We propose a novel approach focused solely on what the Neural Networks learn (i.e., the weights of the network), freeing our method from the pitfalls of possible conflated biases in the considered datasets used for inference. The main contributions of this work can be summarized as:

- We propose IFBiD, a novel bias detector trained with weights of biased and unbiased learned models.
- We analyze how bias is encoded in the weights[1] of deep networks through two different *Case Studies* in image recognition: *A*) digit classification, and *B*) gender detection from face biometrics.
- Our results demonstrate that bias can be detected in the learned weights of Neural Networks. This work opens a new research line to improve the transparency of these algorithms.
- We present two novel databases composed by 84K models trained with different types of biases. These databases are unique in the field and can be used to further research on bias analysis in machine learning.

Related Works

To the best of our knowledge there are no prior works that have attempted to detect the bias of a network by modeling it from learned weights. Existing literature in bias analysis focuses on the performance (outcome) (Bolukbasi et al. 2016; Alvi, Zisserman, and Nellåker 2018; Geirhos et al. 2019; Chen et al. 2019) and those focused on learned representations are few (Stock and Cisse 2018; Serna et al. 2021).

Bias Explainability

There is significant work on understanding neural networks learning processes, which has been useful for diagnosing CNN representations to gain a deep understanding of the biased features encoded in a CNN. In general, they map an abstract concept (e.g. a predicted class) into a domain that the human can make sense of, e.g. images or text; or they collect features of the interpretable domain that have contributed for a given example to produce a decision.

When an attribute often appears alongside other specific visual features in training images, the CNN may use these features to represent the attribute. Thus, features that appear together, but are not semantically related to the target attribute, are considered biased representations. Zhang, Wang, and Zhu (2018) presented a method to discover such potentially biased representations of a CNN.

[1]We used the terms parameters and weights indistinctly to refer the learned filters of a Neural Network.

Nagpal et al. (2019) used Class Activation Maps (CAMs (Zhou et al. 2016)) to obtain the most discriminative regions of interest for input face images in deep face recognition models, and observed that activation maps vary significantly across races.

Also, some investigations show how psychology-inspired approaches can help elucidate bias in DNNs. Examples include Ritter et al. (2017); Geirhos et al. (2019), who found that CNNs trained on ImageNet exhibit a strong bias towards recognizing textures rather than shapes or color. This contrasted sharply with evidence from human behavior, and revealed fundamentally different classification strategies between CNNs and humans.

Bias Detection

Research on bias analysis focuses largely on detecting causal connections between attributes in the input data and outcomes of the learned models (Balakrishnan et al. 2021). This kind of research relies primarily on observational studies where the main conclusions are drawn from benchmarking the learned models. However, in real life applications, it is highly difficult to measure the impact of different covariates on the outcome of a learned model (i.e., it is necessary to demonstrate that correlation implies causation). Balakrishnan et al. (2021) proposed the use of Generative Models to develop causal benchmarks applied to face analysis algorithms. These Generative Models allow manipulation of attributes in the input data, but as the authors mentioned, the synthesis methods are far from being fully controllable and there are still hidden confounders to be considered in these benchmarks.

Stock and Cisse (2018) used an adversarial example approach to model critique (Kim, Khanna, and Koyejo 2016) by feeding the model with a carefully hand-selected subset of examples to subsequently determine whether or not it is biased. Schaaf et al. (2021) introduced different metrics to reliably measure several attribution maps' techniques (Grad-CAM, Score-CAM, Integrated Gradients, and LRP-ϵ) capability to detect data biases. Glüge et al. (2020) attempted to quantify racial bias by clustering the embeddings obtained from the model, but observed no correlation between separation in embedding space and bias.

Our work goes beyond proposals that seek to model bias through the observation of the model outcome in response to particular inputs. Adebayo et al. (2018) already reported the inconsistency of some widely deployed saliency methods, as they are not independent of the data. The present work follows a similar strategy to Serna et al. (2021), who uses the information learned from the model to discover bias by observing the activation of neurons to particular attributes in the inputs. The present work, however, relies solely on the information encoded in the model, without looking at particular input/outputs of the model, thus in an Inference-Free way, with the significant benefits that this represents with respect to all previous works.

The hypothesis behind our proposed Inference-Free Bias Detection (IFBiD) is that bias is encoded in the parameters of a learned model and it can be detected. IFBiD is an interesting and noteworthy effort to contribute to tackling the

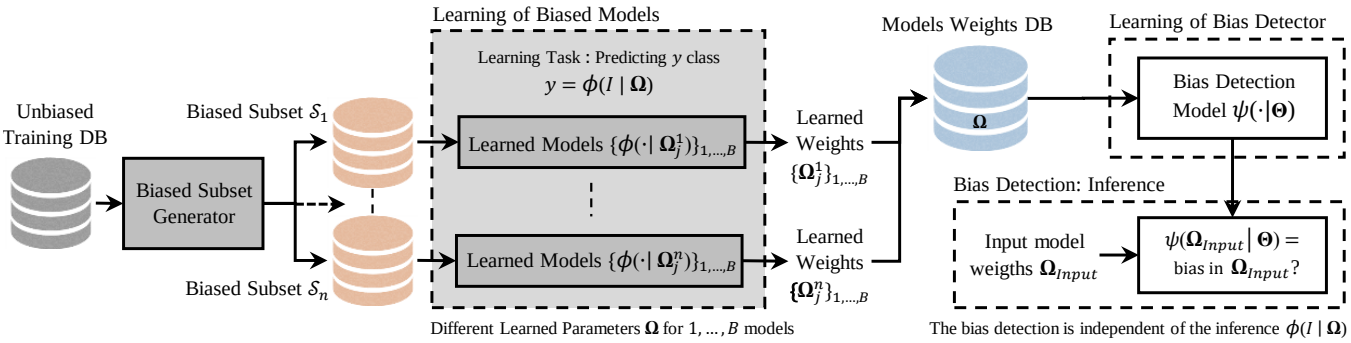

Figure 2: Learning framework of the IFBiD approach based on learned Neural Networks weights.

bias problem. The inference of IFBiD is performed directly over the weights of a learned model, and therefore does not require a causal benchmark based on input/output analysis.

Problem Statement and Proposed Approach

In this work we adopt the formulation proposed by Kleinberg et al. (2019) and Serna et al. (2020) for the discrimination of groups of people, but applicable to any type of bias. The formalization follows:

Definition 1 (Data). $\mathcal{D}$ is a dataset (collection of multiple samples from different classes) used for training and/or evaluating a model $\mathcal{M}$. Samples in $\mathcal{D}$ can be classified according to some criterion d. The set $\mathcal{D}_d^c \subset \mathcal{D}$ represents all the samples corresponding to class c of criterion d.

Definition 2 (Learned Model). The learned model $\mathcal{M}$ is trained according to input data $\mathcal{I} \subset \mathcal{D}$, a Target function T (e.g., digit classification or gender detection), and a learning strategy that maximizes a goodness criterion G on that task (e.g., typically a performance function) based on the output O of the model and the Target function T for the input data $\mathcal{I}$.

Definition 3 (Biased Model). A learned model $\mathcal{M}$ is biased with respect to a specific class c of criterion d if the goodness G on task T when considering the full set of data $\mathcal{D}$ is significantly different to the goodness $G(\mathcal{D}_d^c)$ on the subset of data corresponding to class c of the criterion d.

Typically, as in our case, the model $\mathcal{M}$ is a Neural Network $\phi(\cdot)$, parameterized by $\mathbf{\Omega}$, and the goodness-of-fit criterion consists in minimizing an objective function (e.g. the cross-entropy loss function).

The training process of Neural Networks is usually not deterministic and the resulting parameters $\mathbf{\Omega}$ depend on several elements: training data, learning architecture (e.g., number of layers, number of neurons per layer, etc.), training hyperparameters (e.g., loss function, number of epochs, batch size, learning rate, etc.), initialization parameters, and optimization algorithm.

The existing literature on bias analysis is mainly focused on the inputs $\mathcal{I}$ (Tommasi et al. 2017; Zhang, Wang, and Zhu 2018; Wang, Narayanan, and Russakovsky 2020) and the outputs O to given inputs (Buolamwini and Gebru 2018; Alvi, Zisserman, and Nellåker 2018; Serna et al. 2020). We propose a novel approach to detect bias in the learned parameters $\mathbf{\Omega}$, regardless of the particular input $\mathcal{I}$ or the output $O = \phi(\mathcal{I}|\mathbf{\Omega})$ (see Fig. 2).

IFBiD: Inference-Free Bias Detection Learning

The aim of the bias detection model is to find patterns in $\mathbf{\Omega}$ associated with biased outcomes. We designed the bias detector as a Neural Network $\psi(\cdot)$ represented by its parameters $\mathbf{\Theta}$.

In our approach (detailed in next sections), we train the bias detector using a dataset of biased and unbiased models. The models $\phi(\cdot|\mathbf{\Omega})$ for task T, are biased by training them with biased subsets of the database $\mathcal{S}_1, ..., \mathcal{S}_n \subset \mathcal{D}$. To build a training set for the detector $\psi(\cdot|\mathbf{\Theta})$, we train a number of models $\phi(\cdot|\mathbf{\Omega})$ with each subset S_i, forming $\{\mathbf{\Omega}_j^i\}_{j=1,\dots,B}^{i=1,\dots,n}$ (see Fig. 2), where n is the number of biased subsets and B is the number of models trained with each biased subset. $\mathbf{\Omega}_j^i$ denotes the jth instance of a learned model trained with biased subset i.

Because of the non-deterministic nature of the training process of the network $\phi(\cdot|\mathbf{\Omega})$, the same training subset i is likely to give rise to different $\mathbf{\Omega}$ (i.e., $\mathbf{\Omega}_j^i \neq \mathbf{\Omega}_k^i$). The reason for this is that since the solution space is very large, the solution (which is iteratively approximated) typically arrives at a local minimum that depends on the initialization, the particular training configuration, and the order of the data (LeCun, Bengio, and Hinton 2015). In CNNs, this translates into the fact that filters tend to differ between networks. The bias detector, therefore, has to be able to detect similar filters in different positions and configurations. The problem is analogous to detecting patterns in images, where one can be in different parts of an image.

It is important to underline that the approach requires the target DNNs (i.e., Ω_{Input} in Fig. 2) to have exactly the same architecture as the DNNs used to train the detector. This does not detract from the fact that the task is still challenging, since, as we have explained before, the filters learned by a convolutional network never appear in the same place and are never identical due to the randomness of data presentation and weight initialization (which is, incidentally, the reason why we have also used convolutions in the detector architecture).

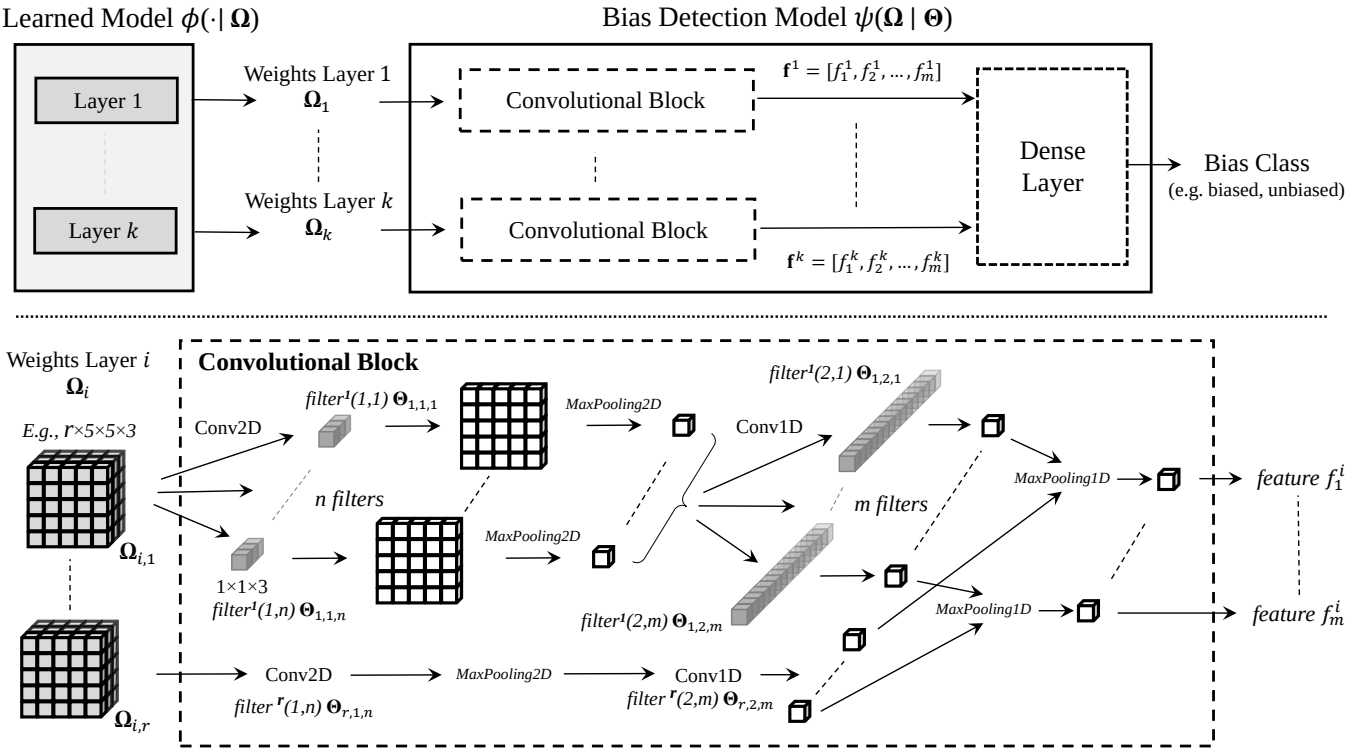

Figure 3: General architecture of a bias detector with the 1×1-conv module variant. The architecture depends on the number of layers k of the model $\mathbf{\Omega}$ to be audited. The depth of the module filters depends on the depth of the input weights. Module variant 1×1-conv consists of the subsequent layers: 1×1 convolution followed by $d \times d$ MaxPooling, then again a one-dimensional convolution with kernel size of 1 followed by a MaxPooling with pool size equal to the number of input filters. Do not confuse the suffixes in this figure (k,i indicates layer) with those in Fig. 2 (j indicates model number).

The Detector

We evaluated many different learning architectures for IF-BiD. The design of the possible architectures has not only taken into account the number and types of layers, but has also depended on the selection of the parameters $\mathbf{\Omega}$ of the model $\phi(\cdot|\mathbf{\Omega})$ used as input for the detector $\psi(\cdot|\mathbf{\Theta})$.

The detector architecture consists of a module to process the weights/filters of each layer, and then a dense layer that concatenates all the outputs of each module. The bias detector architecture consists of multiple modules to process the weights/filters of each layer (thus one module for each layer), and then a fully connected layer that concatenates all the outputs of each module. Fig. 3 shows the general architecture designed for a specific module variant (see below). The components of a module are the same for all layers (convolution, maxpooling, etc.), as well as their order; the only thing that changes are their parameters, which depend on the size of the input weights.

We have developed different approaches, in which the general architecture remains stable, and what changes are the modules. The module variants we analyzed were the following (where $d \times d$ is the dimension of the input filter weights, c is the number of input channels, and r is the number of input filters):

- **MLP**: Flatten $\rightarrow$ Dense(r)
- **1×1 +conv**: Conv2D (1×1) $\rightarrow$ MaxPooling2D ($d \times d$) $\rightarrow$ Conv1D (1) $\rightarrow$ MaxPooling1D (r) $\rightarrow$ Flatten.
- **1×1 +max**: Conv2D (1×1) $\rightarrow$ MaxPooling3D ($d \times d \times k$) $\rightarrow$ Flatten.
- **$1 \times 1 \times 1$ +max**: Conv3D ($1 \times 1 \times 1$) $\rightarrow$ MaxPooling3D ($d \times d \times c$) $\rightarrow$ Flatten.

Convolutions are followed by a relu activation function, and there is always 0.1 dropout afterwards (we have seen that it works best among the values: 0.0, 0.1, 0.2 and 0.3).

Experiments

Datasets of Biased Models

We have created two databases for experimenting in automatic bias detection: DigitWdb and GenderWdb. The databases contain the weights $\mathbf{\Omega}$ of the models $\phi(\cdot, |\mathbf{\Omega})$ used in our experiments for the tasks of digit and gender classification. The databases include 84K models trained with different types of biases (each model has an associated label identifying the bias). These databases are publicly available for further research.[2]

[2]https://github.com/BiDAlab/IFBiD/

Case Study A: Digit Classifier (DigitWdb). We have put together a database that contains the weights $\mathbf{\Omega}$ of 48K digit classification networks $\phi(\cdot, |\mathbf{\Omega})$. For this we have used the colored MNIST database (Kim et al. 2019), which consists of seven replicas of the MNIST database, each with a different level of color bias (of which we have only used four).

To synthesize the color bias, ten different colors were selected and assigned to each digit category as its mean color. Then, for each training image, a color was randomly sampled from the normal distribution of the corresponding mean color, and the digit was colorized. The level of bias of each replica depends on the value of the variance used in the normal distribution: the lower the more bias.

The architecture is the same for all models: a CNN $\phi(\cdot|\mathbf{\Omega})$ with three convolutional layers with relu activation, each followed by a maxpool, and two fully connected layers at the end (with 128 and 10 neurons, a relu and a softmax activation function respectively), with a dropout layer of 0.3 between the two. Each of the trained models results in a total of 50K parameters.

All model parameters $\mathbf{\Omega}$ have been initialized randomly with Glorot uniform (Glorot and Bengio 2010) to avoid possible commonalities. A diagram showing the general construction of a weight database is shown in Fig. 2. The composition is as follows:

- Train: 40K models classified by bias level into four groups, with 10K models per level ($B = 10$K). The models were trained using the first 30K training digits from Colored MNIST. The models have been categorized into four groups depending on the replica subset with which they have been trained ($n = 4$). The level of bias of the replica subset is what determines the level of bias of the model. Groups are: very high bias (color jitter variance of 0.02), high bias (color jitter variance of 0.03), low bias (color jitter variance of 0.04), and very low bias (color jitter variance of 0.05).
- Test: 8K models classified by bias level into four groups (2K models for each level). The models were trained using the last 30K training digits from Colored MNIST and categorized in the same way as the training ones (i.e., from very high bias to very low bias).

Each Colored MNIST biased subset has 60K training digits, so the 30K for train and 30K for test are independent. This means that the DigitWdb models assigned to test have learned with different data than the DigitWdb models assigned to train.

Properties. All models have the same architecture and similar class performance. In this case study, the bias is determined by the color jitter variance of the digit images.

Table 1 shows the average digit classification accuracy of all models trained with the four subsets of different bias levels, from very low bias (subset $\mathcal{S}_1$ with variance of 0.05) to very high bias (subset $\mathcal{S}_4$ with variance of 0.02). The table shows performance on the Colored MNIST's test set, that is, a set of randomly colored numbers, and therefore not biased.

Although not shown, all models exceeded 99% accuracy during training (i.e. in their respective training subsets $\mathcal{S}_i$). This means that our models learn as far as the training set allows. But then, in the (unbiased) test set (i.e., with all digits colored randomly) the number of correct classifications drops considerably. The reason is that the color (present in the training set in a biased way) has been learned as a differentiating element when classifying digits. Thus, the network was not only learning to associate a number to a shape, but also to a color. This is why, subsequently, when finding a digit with a random color, it has much more difficulty in classifying it correctly.

Also, Table 1 shows a clear difference in the performance of the models as a function of the level of bias of the dataset with which it has been trained. This difference is the basis for the experiments carried out in this work.

Table 1: Average digit classification accuracy (in %) in DigitWdb models according to their level of bias. Classification accuracy has been assessed with the Colored MNIST test set, a set of randomly colored numbers, and therefore not biased.

Model Bias	**Digit Classification Accuracy**									
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Very Low	88	94	77	82	90	89	75	84	81	82
Low	79	85	67	69	81	83	65	76	72	69
High	66	76	54	56	72	69	49	64	58	46
Very High	49	51	42	38	59	40	40	51	43	32

Table 2: Average gender classification accuracy (%) in gender classification of all GenderWdb models, according to their class bias. Models are trained with DiveFace dataset.

Model Bias	**Gender Classification Accuracy**		
	Asian	**Black**	**Caucasian**
Asian	89.5%	81.5%	82.9%
Black	82.0%	89.4%	83.0%
Caucasian	80.0%	83.2%	89.2%

Case Study B: Gender Classifier (GenderWdb). We have gathered together a database that contains the weights $\mathbf{\Omega}$ of 36K gender classification networks $\phi(\cdot|\mathbf{\Omega})$. We are aware there are more gender categories other than male and female. Since establishing ground-truth genetic sex is not possible, we use gender as a proxy for sex. We use it as a simplified application of a real-life based problem.

To train the gender classification models that constitute this database, we used DiveFace (Morales et al. 2021). DiveFace is a face dataset contaning 24K identities and three images per identity. Identities are evenly distributed according to gender: male and female, and three categories related to ethnic physical characteristics: Asia, African/Indian, and Caucasian.

As in the DigitWdb database, we have separated the data for training into two independent sets of the same size. (This serves to ensure the future independence of the bias detector training and testing.)

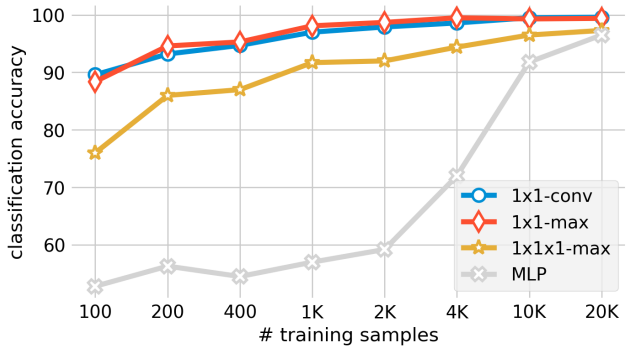

Figure 4: Bias detection accuracy in DigitWdb for the different architectures given the number of training samples (x axis).

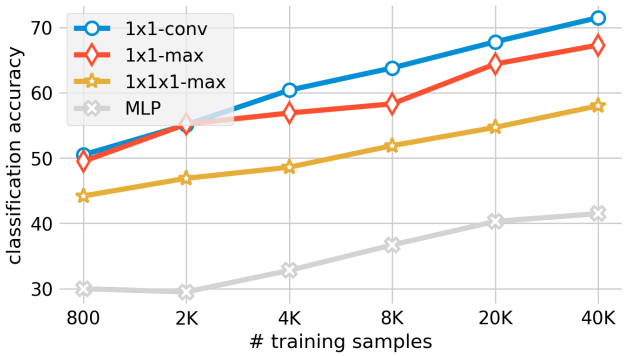

Figure 5: Classification accuracy of the bias level in DigitWdb for the different architectures given the number of training samples (x axis).

We have trained 36K gender classification models $\phi(\cdot|\mathbf{\Omega})$, divided into 30K for training and 6K for testing. The architecture is the same for all models: a CNN with six convolutional layers with relu activation, each followed by a max-pool, and two fully connected layers at the end (with 128 and two neurons, a relu and a softmax activation function respectively). The result is a model with a total of 100K parameters.

All model parameters $\mathbf{\Omega}$ have been initialized randomly with Glorot uniform (Glorot and Bengio 2010) to avoid possible commonalities. The composition is as follows:

- Train: 30K models belonging to three classes of bias ($n = 3$), depending on the subset $\mathcal{S}_i$ with which the model has been trained, with $10K$ models per class ($B =$ 10K). The models were trained using the first 12K faces of each ethnic group of DiveFace: $\mathcal{S}_1$ is asian biased, $\mathcal{S}_2$ is black biased, and $\mathcal{S}_3$ is caucasian biased.
- Test: 6K models with the same three types of bias as train. The models were trained using the last 12K faces of each ethnic group of DiveFace.

Properties. All models have the same architecture and similar class performance. Table 2 shows the average accuracy in gender classification of all models, separated by bias. Bias has been introduced through the subset $\mathcal{S}_i$ with which the model has been trained. In this case study the bias is determined by the ethnicity of the face images. What becomes clear from looking at the table is the strong bias in the performance of the models in each of the groups. Note that ethnicity attributes include the color of the skin, but also more complex anthropomorphic face features.

Results

Bias in digit classification models (Case Study A). First of all we have attempted a binary classification problem: detect strong bias against minimal or no bias.

In this first case we have used models with very high bias and very low bias. Fig. 4 shows the accuracy of bias detection in digit classification models $\phi(\cdot|\mathbf{\Omega})$ for the different architectures given the number of samples the detector $\psi(\cdot|\mathbf{\Theta})$ was trained with. It can be seen that the convolutional architectures show a saturation of classification performance and that it does not take many samples to get great performance. In fact, with the best architectures, 100 training samples are sufficient to achieve a performance of around 90%. These initial results suggest that bias is encoded in the weights $\mathbf{\Omega}$ of the learned models $\phi(\cdot)$ and it can be detected.

A second experiment has been trying to detect the level of bias of a model $\phi(\cdot)$, or in other words, to classify the models according to their level of bias. This is a more complex problem and has required us to test more architectures for the detector $\psi(\cdot)$.

Fig. 5 shows the classification accuracy of the 4 bias levels (cf. initial subsection within Experiments describing the Datasets for Case Study A) for the digit classification models $\phi(\cdot|\mathbf{\Omega})$ and the different architectures given the number of samples with which the detector $\psi(\cdot|\mathbf{\Theta})$ was trained. We see that distinguishing the level of bias in digit classification models is more complicated than simply stating bias-no bias, and that in this case the maximum success rate we achieve in the classification is 70% (note that random chance is 25% for this task). Another important thing to note is the tendency (of good architectures) to keep improving as the training set is increased, they do not seem to be reaching their performance limit.

Bias in gender classification models (Case Study B). After seeing positive results, we made the leap to a more complex problem (i.e., more covariates): detecting ethnic bias in models trained for gender recognition.

Fig. 6 shows the bias classification accuracy of the biased gender recognition models $\phi(\cdot)$ for the different architectures and the number of samples used for training.

The curves show that after a certain number of training samples, the accuracy is no longer increasing in the model with more parameters (that containing Conv3D: $1 \times 1 \times 1$-max). However, it can be seen that when trained with little data, it performs similarly to the rest. The hypothesis that best seems to explain this behavior is that, since there are so many parameters $\mathbf{\Theta}$, the solution space is so large that the choice of a better architectural configuration occurs

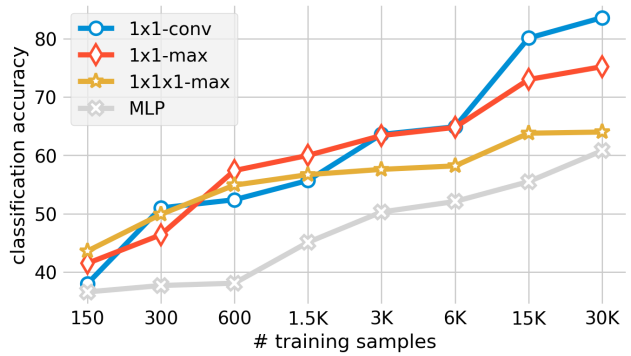

Figure 6: Bias classification accuracy in GenderWdb, for the different architectures given the number of training samples (x axis). Bias is classified into three different categories according to a ethno-demographic criterion, namely Asian, Black, and Caucasian.

automatically, leaving unnecessary parameters unchanged, as if they were not present (Schmidt, Kraaijveld, and Duin 1992). With little training data the model $\psi(\cdot|\mathbf{\Theta})$ adjusts very quickly to those data (losses are practically nil) and in just a couple of epochs it no longer needs to adjust those weights $\mathbf{\Theta}$. On the other hand, when the number of training samples increases, it needs to modify more parameters in order to correlate the training data well, thus losing the generalization capability equivalent to architectures with fewer parameters.

The gender classifier $\phi(\cdot)$ has more layers than the digit classifier $\phi(\cdot)$, twice as many. So the bias detection network $\psi(\cdot)$ for these models has more parameters, and thus the performance is different. The best performance is obtained with the same architecture that also obtains the best performance in the digit models, the architecture with two convolutions: 1×1-conv; reaching 83% detection accuracy. The improvement in the MLP, that has the most parameters, seems to be growing steadily. The rest of the architectures seems that from 15k training samples onwards is when doubling the samples does not increase its performance so much. But it would be necessary to keep doubling the number to check if the trend holds.

We have dealt with many more architectures that are not worth describing here: using two dense layers at the end, adding a dense layer after each convolution, replacing convolutions with dense layers, playing with dropout, etc.; all resulting in worse performance than the learning architectures reported here.

SOTA Comparison

Table 3 shows the comparison with a recent state-of-the-art bias detection method (Serna et al. 2021), which consists of measuring the activation of the last layer upon image input. We also use an SVM with radial basis function (RBF) kernel as a baseline, trained in the same way as our detector. The table shows the percentage of biased models detected by InsideBias, the RBF SVM, and by our method. The 6K GenderWdb test models were used, being 2K of each type of bias.

Table 3: Percentage of biased models detected according to their type of bias.

Method	**Bias Detection Accuracy**		
	Asian	**Black**	**Caucasian**
RBF SVM	71%	30%	26%
InsideBias (Serna et al. 2021)	23%	86%	3%
IFBiD (ours)	95%	79%	79%

In order to apply InsideBias, we used 60 images from DiveFace, with 20 of each ethnicity and the same number of men and women. Separately, as input to the SVM, we used the parameters of the models put together as a vector of length 97K.

Our method shows considerable superiority: it has a good hit performance on models with all biases, whereas the other methods only detect well a single type of bias in the models.

Conclusion

We presented a novel approach called IFBiD (Inference-Free Bias Detection) to analyze biases in neural networks: by auditing the models through their weights. Our experiments demonstrate the existence of identifiable patterns associated with bias in the weights of a trained Neural Network (Terhorst et al. 2022). We conducted experiments in two computer vision use cases: digit and face gender classification (Serna et al. 2021). This involved generating two databases with thousands of biased models each. The first, DigitWdb, with models trained on the Colored MNIST database (Kim et al. 2019); and the second, GenderWdb, with models trained on a face database, DiveFace (Morales et al. 2021).

We used each database to train bias detectors following the proposed IFBiD principles. We have evaluated a number of architectures and have found that in both cases it is possible to achieve a good performance in bias detection. In the digit classification models we were able to detect whether they presented strong or low bias with more than 99% accuracy, and we were also able to classify between four levels of bias with more than 70% accuracy. For the face models, we achieved 83% accuracy in distinguishing between models biased towards Asian, Black, or Caucasian ethnicity. In both cases the experiments are open-ended in the absence of increasing both databases. This has been evident in the plots with the experiments carried out for different sizes of the training set.

We evaluated our approach by varying the nature of the data (i.e., digits and face images), type of architecture (i.e., number of layers, units), and optimization strategy (e.g., loss function). For future work, the generalization capabilities of our proposed bias detection approach should be studied in more depth. The training process of the biased models used for training IFBiD can be affected by hidden confounders that need to be considered.

Acknowledgments

This work has been supported by projects: TRESPASS-ETN (MSCA-ITN-2019-860813), PRIMA (MSCA-ITN-2019-860315), BIBECA (RTI2018-101248-B-I00 MINECO/FEDER), and BBforTAI (PID2021-127641OB-I00 MICINN/FEDER). I. Serna is supported by a FPI fellowship from UAM.

References

Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31, 9525–9536. Montréal, Canada: Curran Associates Inc.

Alvi, M.; Zisserman, A.; and Nellåker, C. 2018. Turning a Blind Eye: Explicit Removal of Biases and Variation from Deep Neural Network embeddings. In *European Conference on Computer Vision (ECCV)*, 556–572. Munich, Germany.

Balakrishnan, G.; Xiong, Y.; Xia, W.; and Perona, P. 2021. Towards Causal Benchmarking of Bias in Face Analysis Algorithms. In *Deep Learning-Based Face Analytics*, 327–359. Springer.

Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; Chatila, R.; and Herrera, F. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58: 82–115.

Bau, D.; Zhu, J.-Y.; Strobelt, H.; Lapedriza, A.; Zhou, B.; and Torralba, A. 2020. Understanding the Role of Individual Units in a Deep Neural Network. *Proceedings of the National Academy of Sciences*, 1–8.

Bolukbasi, T.; Chang, K.-W.; Zou, J.; Saligrama, V.; and Kalai, A. 2016. Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings. In *Advances in Neural Information Processing Systems*, 4349–4357. Barcelona, Spain.

Buolamwini, J.; and Gebru, T. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In Friedler, S. A.; and Wilson, C., eds., *Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, 77–91. New York, NY, USA.

Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; and Kurtzman, T. 2019. Hidden Bias in the DUD-E Dataset Leads to Misleading Performance of Deep Learning in Structure-based Virtual Screening. *PLOS ONE*, 14(8).

Fierrez, J.; Morales, A.; and Ortega-Garcia, J. 2021. *Encyclopedia of Cryptography, Security and Privacy*, chapter Biometrics Security. Springer.

Geirhos, R.; Rubisch, P.; Michaelis, C.; Bethge, M.; Wichmann, F. A.; and Brendel, W. 2019. ImageNet-trained CNNs are Biased Towards Texture; Increasing Shape Bias Improves Accuracy and Robustness. In *International Conference on Learning Representations (ICLR)*. New Orleans, Louisiana, USA.

Glorot, X.; and Bengio, Y. 2010. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Teh, Y. W.; and Titterington, M., eds., *Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, 249–256. Chia Laguna Resort, Sardinia, Italy: PMLR.

Glüge, S.; Amirian, M.; Flumini, D.; and Stadelmann, T. 2020. How (Not) to Measure Bias in Face Recognition Networks. In *Artificial Neural Networks in Pattern Recognition*, 125–137. Springer International Publishing.

Kim, B.; Khanna, R.; and Koyejo, O. 2016. Examples Are Not Enough, Learn to Criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems (NIPS)*, 2288–2296. Barcelona, Spain.

Kim, B.; Kim, H.; Kim, K.; Kim, S.; and Kim, J. 2019. Learning Not to Learn: Training Deep Neural Networks With Biased Data. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 9012–9020. Las Vegas, Nevada, USA: IEEE.

Kleinberg, J.; Ludwig, J.; Mullainathan, S.; and Sunstein, C. R. 2019. Discrimination in the Age of Algorithms. *Journal of Legal Analysis*, 10: 113–174.

Koh, P. W.; and Liang, P. 2017. Understanding Black-box Predictions via Influence Functions. In *International Conference on Machine Learning (ICML)*, volume 70, 1885–1894. PMLR.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep Learning. *Nature*, 521(7553): 436–444.

Mahendran, A.; and Vedaldi, A. 2015. Understanding Deep Image Representations by Inverting Them. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 5188–5196. Boston, MA, USA: IEEE.

Montavon, G.; Samek, W.; and Müller, K.-R. 2018. Methods for Interpreting and Understanding Deep Neural Networks. *Digital Signal Processing*, 73.

Morales, A.; Fierrez, J.; Vera-Rodriguez, R.; and Tolosana, R. 2021. SensitiveNets: Learning Agnostic Representations with Application to Face Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 43(6): 2158–2164.

Nagpal, S.; Singh, M.; Singh, R.; Vatsa, M.; and Ratha, N. 2019. Deep Learning for Face Recognition: Pride or Prejudiced? *arXiv:1904.01219*, 1–10.

Ntoutsi, E.; Fafalios, P.; Gadiraju, U.; Iosifidis, V.; Nejdl, W.; Vidal, M.-E.; Ruggieri, S.; Turini, F.; Papadopoulos, S.; Krasanakis, E.; et al. 2020. Bias in Data-driven Artificial Intelligence Systems—An Introductory Survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(3).

Ortega, A.; Fierrez, J.; Morales, A.; Wang, Z.; and Ribeiro, T. 2021. Symbolic AI for XAI: Evaluating LFIT Inductive Programming for Fair and Explainable Automatic Recruitment. In *IEEE/CVF Winter Conf. on Applications of Computer Vision Workshops (WACVw)*.

Ritter, S.; Barrett, D. G.; Santoro, A.; and Botvinick, M. M. 2017. Cognitive Psychology for Deep Neural Networks: A Shape Bias Case Study. In *International Conference on*

Machine Learning (ICML), volume 70, 2940–2949. Sydney, NSW, Australia: PMLR.

Schaaf, N.; de Mitri, O.; Kim, H. B.; Windberger, A.; and Huber, M. F. 2021. Towards Measuring Bias in Image Classification. *arXiv preprint arXiv:2107.00360*.

Schmidt, W. F.; Kraaijveld, M. A.; and Duin, R. P. 1992. Feed Forward Neural Networks with Random Weights. In *International Conference on Pattern Recognition (ICPR)*. The Hague, Netherlands: IEEE Computer Society.

Serna, I.; Morales, A.; Fierrez, J.; Cebrian, N., M. Obradovich; and Rahwan, I. 2020. Algorithmic Discrimination: Formulation and Exploration in Deep Learning-based Face Biometrics. In *AAAI Workshop on Artificial Intelligence Safety (SafeAI)*. New York, NY, USA.

Serna, I.; Peña, A.; Morales, A.; and Fierrez, J. 2021. InsideBias: Measuring Bias in Deep Networks and Application to Face Gender Biometrics. In *IAPR Intl. Conf. on Pattern Recognition (ICPR)*, 3720–3727. IEEE.

Stock, P.; and Cisse, M. 2018. ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases. In *European Conference on Computer Vision (ECCV)*, 498–512. Springer International Publishing.

Stone, P.; Brooks, R.; Brynjolfsson, E.; Calo, R.; Etzioni, O.; Hager, G.; Hirschberg, J.; Kalyanakrishnan, S.; Kamar, E.; Kraus, S.; et al. 2016. Artificial Intelligence and Life in 2030. *One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel*, 52.

Terhorst, P.; Kolf, J. N.; Huber, M.; Kirchbuchner, F.; Damer, N.; Morales, A.; Fierrez, J.; and Kuijper, A. 2022. A Comprehensive Study on Face Recognition Biases Beyond Demographics. *IEEE Trans. on Technology and Society*.

Tommasi, T.; Patricia, N.; Caputo, B.; and Tuytelaars, T. 2017. *A Deeper Look at Dataset Bias*, 37–55. Springer International Publishing.

Wang, A.; Narayanan, A.; and Russakovsky, O. 2020. REVISE: A Tool for Measuring and Mitigating Bias in Visual Datasets. In *European Conference on Computer Vision*, 733–751. Springer International Publishing.

Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; and Lipson, H. 2015. Understanding Neural Networks Through Deep Visualization. In *Intenational Conference on Machine Learning (ICML) Deep Learning Workshop*. Lille, France.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision (ECCV)*, 818–833. Zurich, Switzerland: Springer.

Zhang, B. H.; Lemoine, B.; and Mitchell, M. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 335–340.

Zhang, Q.; Wang, W.; and Zhu, S.-C. 2018. Examining CNN Representations with Respect to Dataset Bias. In *AAAI Conference on Artificial Intelligence*, volume 32 of *AAAI'18*. AAAI Press.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2921–2929. Las Vegas, Nevada, USA: IEEE.