Unravelling a Paper on Algorithmic Bias Through Reconstruction and Reimplemantation

Lucas Fijen (10813268)
lucas.fijen@gmail.com
University of Amsterdam

Dante Niewenhuis (11058595)
d.niewenhuis@hotmail.com
University of Amsterdam

Jonathan Mitnik (10911197)
jmitnik@gmail.com
University of Amsterdam

Pieter de Marez Oyens (10002403)
oyenspieter@gmail.com
University of Amsterdam

TA: Simon Passenheim
simon.passenheim@googlemail.com
University of Amsterdam

ABSTRACT

As fairness becomes an increasingly important topic in the field of AI, the abundancy of available algorithms and methods also grows. This paper tries to replicate a proposed novel method from the paper *Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure* by Amini et al., in which facial recognition systems are debiased.

The paper's conclusion and results were successfully reproduced, but parameter settings and techniques were inadequately explained. Unofficial code was found and used to correctly implement lacking information. Furthermore the authors made slight errors in the use of evaluation metrics and the the evaluation dataset could have been better chosen.

1 INTRODUCTION

As AI is applied on a broader societal level, fairness is becoming an increasingly more important topic. Fairness can be defined in two parts: individual fairness, meaning that similar individuals should be treated equally, and group fairness, stating the each group of individuals should be treated comparably [4]. An unfair AI model can thus be described as a model that makes more mistakes or favors certain individuals or groups of people. Unfair algorithms make misclassifications that can lead to innocent albeit racist results [5] or potentially physical harmful behaviour [12].

A few examples of algorithmic decision making applied in society are: predicting prison-sentence lengths [3], criminal-profiling [6] and deciding loan-applications [8]. It is with these and other cases in mind that we can say it is pertinent that these algorithms are fair and unbiased.

Different types of bias can occur at any moment in the machine learning pipeline. Main types of bias include historical-, representational- and measurement-bias [11]. Out of these three, researchers can intervene in the last two types. Algorithms that counter unfairness are called mitigation algorithms and can be preformed in the following pipeline-steps: data pre-processing before training, in-processing during training and post-processing after training.

Admittedly data is one of the biggest sources of unfairness [2] and thus research has focused on mitigating this. Existing techniques include creating artificially debiased data or resampling techniques among many other options. This paper attempts to recreate one of these techniques, namely the technique developed by Amini et al. as described in their paper *'Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure'*. This paper will be referred to as the *original paper*.

The authors propose a debiasing algorithm that actively debiases facial image data. The final goal is facial detection with equal correct classifications for different types of faces (e.g. light and dark skin, male and female). The authors integrate debiasing of data directly into the training process of a model and use the learning of the underlying latent structure of images to help define useful features of the images.

The original paper proposes using the underlying latent variables to adjust sampling probabilities of individual data points while training and creating a semi-supervised model that learns debiased face classification. The original paper's the method is evaluated on test data that includes labels on different races and genders.

The paper was not accompanied by official code. However, an unofficial code repository [1] created by one of the main authors was found. This repository was used to fill in implementary gaps of information when the original paper did not provide enough details. This repository was not cited in the original paper, and as such can not be considered to be an official implementation. Nevertheless, this repository is to be considered the closest additional feedback, and is from here on referred to as the *original code*.

The main goal of this paper is to investigate whether the results of the original paper can be reproduced as well as point out possible flaws and points of improvement.

2 METHOD

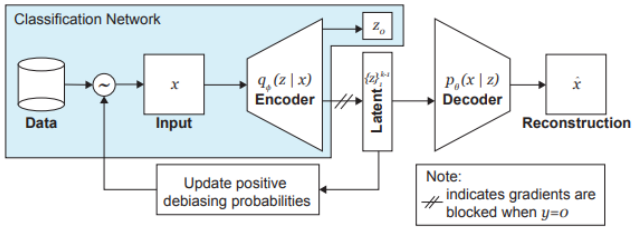
The original paper describes a model that combines a classification network with a Variational Auto Encoder (VAE). On top of those two networks it introduces a sampling method for the creation of minibatches based on latent biases in the training-dataset.

2.1 Original Model

The introduced model in the original paper combines a classification network with a VAE as can be seen in Figure 1. Whereas a normal VAE encoder has 2k outputs z (μ and σ for k latent variables), the combined encoder has 1 extra output variable (z_0) that describes the classification of faces. On this output a sigmoid function is applied to map the predictions to a range between 0-1.

[1] https://github.com/aamini/introtodeeplearning/blob/master/lab2/solutions/Part2_Debiasing_Solution.ipynb

Figure 1: The architecture of the model used in this. Image taken from the original paper [1, Figure 2]



The decoder of the VAE network is standard, so the distributions of the latent space are sampled using the reparameterisation trick, and those samples are used as input for a decoder network that reconstructs the images. In this network there are three loss functions: A cross-entropy loss on the binary classification output, an MSE (VAE)loss on the reconstructed images and a KL divergence (VAE) loss on the latent distributions which regularises the latent space. The two VAE losses are only calculated for positive samples (faces), and the cross entropy loss is calculated on all the input images. By training the loss of the VAE on faces only, the latent space is forced to represent latent features of faces.

The resulting total loss function can be seen in Equation 1, where c1, c2 and c3 can be used to weight the different losses.

$$\mathcal{L}_{TOTAL} = c_1 \left[y \log\left(\frac{1}{\hat{y}}\right) + (1-y)\log\left(\frac{1}{1-\hat{y}}\right)\right] + \mathbb{1}_{y=1}\left(c_2\left[\parallel x - \hat{x} \parallel_2\right] + c3\left[\frac{1}{2}\sum_{j=1}^{k}(\sigma_j + \mu_j^2 - 1 - \log(\sigma_j))\right]\right) \tag{1}$$

2.2 Debiasing

In the original paper a new method is introduced to compensate for biases in the training data throughout the creation of adapted minibatches. The idea behind this method is that latent biases in a dataset are also represented in the latent space of the VAE (e.g. skin color, gender). Images that are mapped to positions in the latent space that are less often seen in the full training set, are expected to represent an under-represented group in the dataset. The intuition of the debiasing algorithm is to give those images which are mapped to a underrepresented area of the latent space a higher probability to end up in minibatches.

To calculate the mapping towards the latent space, at the beginning of each epoch all the images of faces in the dataset are forwarded through the encoder to observe the latent space. This provides us an approximation of the latent distribution $\hat{Q}(z|X)$. $\hat{Q}(z|X)$ can be used to determine how rare the latent distribution from a given data point is. To account for the complexity of high-dimensional latent spaces, $\hat{Q}(z|X)$ is calculated using the histograms of each latent variable $\hat{Q}_i(z|X)$. These histograms are then used to calculate the relative sampling probability $\mathcal{W}$ for each image using the following formula:

$$\mathcal{W}\left(z\left(\mathbf{x}\right)|X\right) \propto \prod_i \frac{1}{\hat{Q}_i\left(z|X\right) + \alpha} \tag{2}$$

In this formula α is a variable that is used regulate the amount of debiasing in an algorithm. However, this proposed formula is unstable when dealing with high-dimensional latent space. If non-normalized $\hat{Q}_i(z|X)$ values are used, each $\mathcal{W}\left(z\left(\mathbf{x}\right)|X\right)$ will result in underflow and thus result in a probability of zero. When $\hat{Q}_i(z|X)$ are normalized, each $\mathcal{W}\left(z\left(\mathbf{x}\right)|X\right)$ will result in overflow and thus result in a probability of infinity.

2.3 Max Probability Selection

Two main methods were explored in this paper to solve the problem of the unstable formula for debiasing provided in the original paper. The first method was found in the original code. The author solved the problem by taking the highest resulting value of the inverse of Q + alpha. In this paper this method will be referred to as the *max* debias method. While this method does explore the latent space, it disregards all latent variables beside the one with the highest probability. Therefore this method is not proportional to the originally given Equation 2.

The second method of debiasing is an extension of the max debias method and will be referred to as the *max5* debias method. This method is similar to the max debias method but instead of taking only the highest value, it takes the product of the highest 5 values which can lead to a better approximation of the latent space. As this method takes more dimensions of the latent space into account, it is a closer approach towards the proportional to property of the original Equation 2 compared to the *max* debias method.

2.4 Creation of Histograms

Another part that was left to interpretation was the initialisation of the histograms. In the original code it seems that only μ's of the latent variables are used, which is also the primary used method in this paper.

Another method for constructing the histograms from the actual Gaussian functions is newly introduced in this paper. This method will be referred to as Gaussian debiasing method. In this method the histograms per latent dimension are defined by summing up the values of the Gaussian functions described in the latent space over a linspace of 100 data points. Then the resulting histograms are multiplied with each value of the distribution function of an individual images' latent space, resulting in a more accurate histogram built from the original distribution functions, rather than just their mu. This last method is however computationally more expensive as the actual distribution functions have to be calculated.

3 EXPERIMENTAL SETUP

The primary goal of this project was to reproduce the original paper as accurately as possible. Besides recreating the original setup, we also explored other methods that could improve performance or give clearer results.

3.1 Metrics Used

This section is included to clarify used metrics. The original paper uses metrics which are, in our opinion, not correctly named. During evaluation three different metrics are used: Precision, Recall and Accuracy, as explained by Joshi [7]. In this paper, images containing faces are considered the positive class and non-faces the negative class.

3.2 Datasets

During training and evaluation three different datasets are used. Training of the models was done using the CelebA[2] and the ImageNet[3] datasets. The CelebA dataset consists of faces of celebrities which are more or less centered. ImageNet images are random images without human faces in them, sampled from various categories. The original paper described the use of 200.000 images from both the faces and non-faces datasets during training. Contrasting the original code that used approximately 55.000 images for both faces and non-faces. The CelebA images were cropped and adjusted so the faces are centered and the background is virtually non-existent.

For our implementation, we used the data found in the original code, as the boxes for the raw CelebA data were not precise enough and the datastructure being computationally heavier. Additionally all images used in this paper are resized to sizes of 64x64 if necessary.

For evaluation the Pilot Parliaments Benchmark (PPB) dataset is used. The PPB dataset consists of 1270 images of Parliament members from six countries. Each image contains corresponding metadata detailing the gender and skin color of its subject, which makes it possible to filter the dataset on these factors.

For an example of what these look like, see Figure 7 in the appendix.

3.3 Original Setup

The original setup states the following problem: given a set of paired training data samples $\mathcal{D}_{train} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}$, were $\mathbf{x}^{(i)}$ are raw pixel values of images, and $\mathbf{y}^{(i)}$ are labels indicating if the image is a face or not. The goal is to use the de-biasing method to train a model that is more fair and unbiased. A fairer model is described as a model were the difference in classification performance between different groups within the dataset is lower. In the original paper those groups were based on skin color and gender.

The model used for the given problem is the DB-VAE explained in section 2 and shown in Figure 1. The encoder is a convolutional neural network consisting of four sequential convolutional layer with filter size 5x5 and 2x2 stride. The convolutional layers use LeakyReLU activations with a slope of 0.2 and batch normalization. The original paper uses normal ReLU but we found our models to be more stable when using LeakyReLU. The final classification is done using two fully connected layers resulting in a size of $2k + 1$ neurons, with k being the number of latent variables. All the trained models had a latent space of 100, which was not specifically mentioned in the paper but was gathered from the original code. The decoder mirrors the encoder resulting in reconstructed images of size 64x64. To result in 64x64, output padding was used, which results in gray lines around the reconstructed images.

In total 13 different models were trained: for the max, max5 and Gaussian debiasing method four models each were trained with varying degrees of debiasing, defined as α. The losses of the debiased models are calculated using Equation 1 with $c_3 = 0.1$, which was gathered from the original code. A reference model was trained using no debiasing. All models were trained for 50 epochs, and were trained 5 times from scratch to ensure stable results.

Evaluation of the models is done using the PPB dataset. For the evaluation, patches from each image are extracted using sliding windows of varying dimensions, as can be seen in Figure 6 in the appendix. The original paper did not specify this process further but in our implementation for each image around 1000 patches are extracted. All patches of the image are classified and an image is classified as a face if any of the patches is classified as a face by the model. The metric used for the quality of the model is given by the percentage of faces classified correctly by the model. In the original paper this metric is called accuracy but we think that recall is a much more suitable name since only positive samples are used for evaluation (see subsection 3.1).

To compare the bias of a model, the PPB dataset is separated into four subsets: dark male, dark female, light male and light female. A model is less biased than another if the variance of the recall across the four subsets is lower.

3.4 Extensions

In the original paper the models were evaluated only on positive samples. We feel that this method of evaluation only does half of the necessary steps for proper evaluation. It completely disregards the models ability to classify images of non-faces, which can lead to results that seem much better than they actually are and can thus be misleading. To combat this problem we also evaluated the models on images of non-faces using the same method of extracting patches as used on the PPB dataset. From these result we can calculate the precision and the accuracy of the model which will give a more complete overview of its capabilities and possible problems.

4 RESULTS

The following section will discuss the results gathered from the evaluation methods discussed in subsection 3.3 and subsection 3.4. Further implications of the gathered results will be discussed in section 5 and section 6.

4.1 Max Debias

The results achieved using the max debias method are very promising (Figure 2b). The biggest improvement of debiased model was made in the dark male group, increasing from an average of 85% recall in the non-debiased model to an average of 92% in the best debiased model. Three groups did not increase as strong but still showed improvement. The variance of the trained models decreased significantly from an average of 33 from the non-debiased model to an average of 17 from the best debiased model. The results of the max debias method are similar to the results shown in the original paper (Figure 2), and can be considered to satisfy the reproduction of the main results of the original paper.

[2] http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
[3] http://www.image-net.org/about-overview

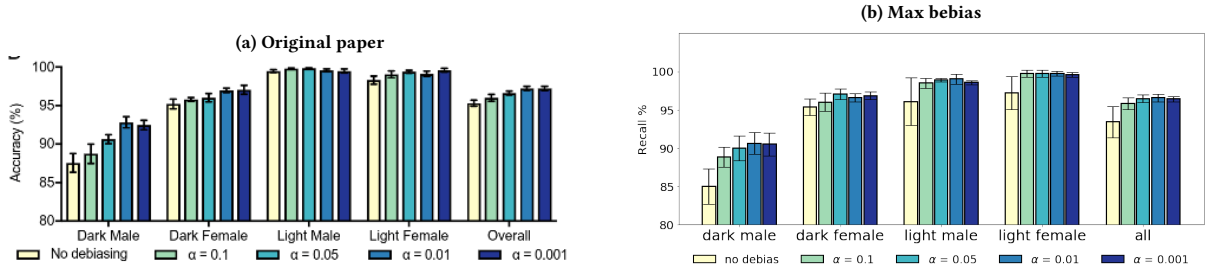


Figure 2: Bar plot of the recall of the different models trained using different degrees of debiasing. Plot (a) shows the results from the original paper. Plot (b) shows the results from the models trained using the Max debiasing method. The results look very similar.

4.2 Max5 and Gaussian Debias

Both the max5 and the Gaussian debias methods were showing some promising results but were also more unstable than the max method (Figure 4). While both methods were able to get very good models, the spread was also much higher (Figure 5). This makes both methods not recommended to use.

4.3 Precision, Recall and Accuracy

Figure 3 expands on the original paper's use of recall by also showing the various precision and accuracy values corresponding to the alpha-values. These values portray the general performance of the model, and how the debiasing influences its overall ability to classify a face according to the evaluation technique. As noted in subsection 4.1, the recall generally increases when debiasing is used. However, it is important to notice that in contrast the precision decreases sharply when debiasing is applied. The true accuracy also shows a decrease in general performance.

5 DISCUSSION

5.1 Recall, Precision and Accuracy

As described in section 4, by using specific experimental parameters, the results presented by the original paper were approximated with success. Compared to the standard non-debiased classifier, the debiased sampler has indeed a lower variance in accuracy scores.

However, the original papers' evaluation method only compared the abilities of debiased and non-debiased models to the recall on faces, ignoring precision and incorrectly referring to this metric as accuracy. By inappropriately referring to the main metric as accuracy instead of recall, the original papers' authors have misleadingly ignored an important potential drawback of the model's performance when introducing debiasing. This is supported in Figure 3, as the precision decreases more strongly than the recall increases. As a result, the model might be able to recognise faces better for different skin-colors, but less able to distinguish faces from non-faces.

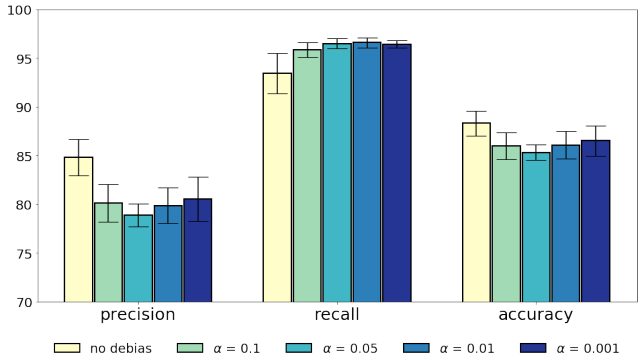
An arbitrary example why this might be a problem is that a model may learn to be biased to always find a face in any part of an image, regardless of whether a face is actually present. If this metric would be withheld, similar results could be presented for any skin-color, but the model would not have been more effective in general. Nevertheless, this report maintains the idea that the decrease in precision is justified within the scope of fairness in AI and while it is important to report such drawbacks, it should not completely negate the conclusion of the original paper.

As to why the decrease in precision occurs; it could be caused by noise from the dataset itself. The bias between images is not solely due to a subject's sensitive features in an image, but also can be influenced by challenging lighting, angle and various facial decorations. A network is inclined to be more challenged by these traits and as such, learn to sample these traits more often. By training more on these different features as well, images with similar attributes may be misidentified as faces as well. One suggestion we present to possibly increase precision is by filling the non-faces dataset with faces with facial decoration, and teach a model that these traits are not relevant features.

5.2 Confounding Variables in Evaluation

Aside from incorrectly noting the appropriate metrics, this report argues that the evaluation data-set used in the original paper seems to contain more confounding variables. The PPB dataset shows a great disparity in quality and noisy photos between faces from people of African descent and those of European descent. These variables could cause the network to possibly fire more easily on high-quality facial profiles than lower-quality ones. Evaluating the

Figure 3: Bar plot of the recall, precision and accuracy of different models. The models were trained using the max debias method.



network on a data-set with a more equal distribution within the color-classes would help combat this implicit bias in the evaluation set.

5.3 Reproducibility

In terms of reproducibility, the paper lacked a number of essential parameter settings necessary to receive the retrieved results. This lacking brought various problems, varying from obstructing to crucial, as certain parameters settings lead to completely different conclusions.

Parameters settings that were not specified in the original paper are the three constant parameters c_1, c_2, c_3 for the three loss terms (Equation 1) and instead were extract from the original code. The different methods for calculating the histogram probability distribution would cause a numerical over- or underflow if followed strictly and as such, as described in subsection 2.4, various methods had to be tested including the one found in the original code. While the paper and the repository manage to reach the same results, had it not been for the knowledge to use the max probability as was found in the main paper, the results could have been completely different.

Another crucial set of hyper-parameters which were neglected, were possible window sizes used for evaluation along with the stride-steps. While using a relatively high stride seems irrelevant, a stride that is too high will nullify the original papers' conclusion, which shows that the evaluation method needs to be explained more thoroughly in the original paper to prevent an error in reproducibility. Other missing hyper-parameters include the latent space size, the number of samples taken to build a histogram and the amount of bins of these histograms.

6 BROADER IMPLICATIONS

Even though fairness is an important factor in model creation, it is not the only one. Other factors such as *accountability, confidentiality* and *transparency* need to be taken into account when creating models that can confidently be deployed. In this section will we discuss shortly the implications the aforementioned subjects have on the replicated model and how the current model can be improved taking these factors into account. However, in this particular case, the model does not use privacy sensitive data and as such confidentiality is a none issue and won't be included it in this discussion.

6.1 Accountability

In Raji and Buolamwini's paper in-production facial recognition models are audited and their bias is researched. The companies who have these models in production are notified on the results on the audits before they are released to the general public. The paper compares the model performance for each audited company before and after the release of the initial results, and compares these change to model performance on unaudited companies. The research shows that companies who are audited and take the flaws of biased-models seriously, show great performance increase over a matter of months, while unaudited companies' models do not perform as good as the improved models of audited companies. The report reflects upon system like the one implemented in this paper and the companies who use them seem satisfied with performance even if they are biased, unless they are called out on it. It calls for better models that are inherently as un-biased as possible.

6.2 Transparency

Why specific models classify certain images as, for example, face or non-face, can be hard to interpret, especially when using deep neural-networks. Ribeiro et al. propose a method to locally interpret models using a technique called LIME. In their paper [10] they give an example with deep networks classifying images, and turning certain pixels *on* which, to the model, are important, and otherwise *off*. The implemented model could benefit from this technique in two-ways, one of which is in the first part of the network where images are either classified as face or non-face. Using this technique sensible choices for models can be made if the the model shows it takes the right pixels into account when trying to identify if an image as a face or not. Secondly, the second part of the network where the images are classified on gender and race, it can be useful to see if the latent space correlates with the pixels shown by LIME. For example, men usually have larger jawbones and heavier brows, and a good model may want to take those features and pixels more into account than others. This technique, together with the representation of the discovered latent space can make the model highly transparent and easily explainable. When dealing with such a delicate subject such as involuntary bias, it is important for researches and production team to explain why a model is biased if it is, and it may help future studies to prevent this.

7 CONCLUSION

The goal of this paper was to reproduce a facial recognition debiasing method from the paper *Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure* by Amini et al.. Parameter values of the original paper were not clearly documented and instead derived from the original code. Critical formulas to calculate histogram probabilities were also not clearly defined with over- and underflow problems as a result. Due to this, the original paper can not be considered completely reproducible by its own merit. However, unofficial code was found written by the main author of the paper from which hyper-parameters as well as formulas were used. This made it possible for us to achieve similar results as the original paper.

Furthermore the original authors seemed to have misused evaluation metrics. We argued that the *accuracy* metric is actually the *recall* metric. The authors also left out other important metrics such as precision. We have shown that the implemented method does increase in recall and reduces variance but this comes at the cost of the models precision and accuracy. However, the debiased version does perform well when looking at recall which in the context of fairness is the goal of the model.

REFERENCES

[1] Alexander Amini, Ava Soleimany, Wilko Schwarting, Sangeeta Bhatia, and Daniela Rus. 2019. Uncovering and Mitigating Algorithmic Bias through Learned Latent Structure. (2019).

[2] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[3] Richard A Berk, Susan B Sorenson, and Geoffrey Barnes. 2016. Forecasting domestic violence: A machine learning approach to help inform arraignment decisions. *Journal of Empirical Legal Studies* 13, 1 (2016), 94–115.

[4] Maarten de Rijke. 2020. FACT Lecture Notes.

[5] Conor Dougherty. [n.d.]. Google Photos Mistakenly Labels Black People 'Gorillas'. *The New York Times* ([n. d.]). https://bits.blogs.nytimes.com/2015/07/01/google-photos-mistakenly-labels-black-people-gorillas/

[6] David J Icove. 1986. Automated crime profiling. *FBI L. Enforcement Bull.* 55 (1986), 27.

[7] Renuka Joshi. 2016. Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. [Online; accessed 28-Januari-2020].

[8] Amitabha Mukerjee, Rita Biswas, Kalyanmoy Deb, and Amrit P Mathur. 2002. Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management. *International Transactions in operational research* 9, 5 (2002), 583–597.

[9] Inioluwa Deborah Raji and Joy Buolamwini. 2019. Actionable auditing: Investigating the impact of publicly naming biased performance results of commercial ai products. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 429–435.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[11] Harini Suresh and John V Guttag. 2019. A Framework for Understanding Unintended Consequences of Machine Learning. *arXiv preprint arXiv:1901.10002* (2019).

[12] Benjamin Wilson, Judy Hoffman, and Jamie Morgenstern. 2019. Predictive inequity in object detection. *arXiv preprint arXiv:1902.11097* (2019).

A CONTRIBUTIONS

Equal contribution. Listing order is random. Dante and Lucas provided most of the VAE code due to their extensive experience with this topic. Jonathan provided most of the data loader code while Pieter aided in this. Models where evaluated and documented by different team members, while the writing of the initial paper was overseen by Pieter. All team members contributed to individual parts. Pieter writing mostly the abstract, introduction, broader implication and conclusion. Jonathan the discussion, Lucas and Dante the original and proprietary setups. All team members then scrumptiously rewrote the whole papers together. The finishing touches like the API, doc strings, comments and cleaning up the code repository was a task shared and loathed by all.

B FIGURES

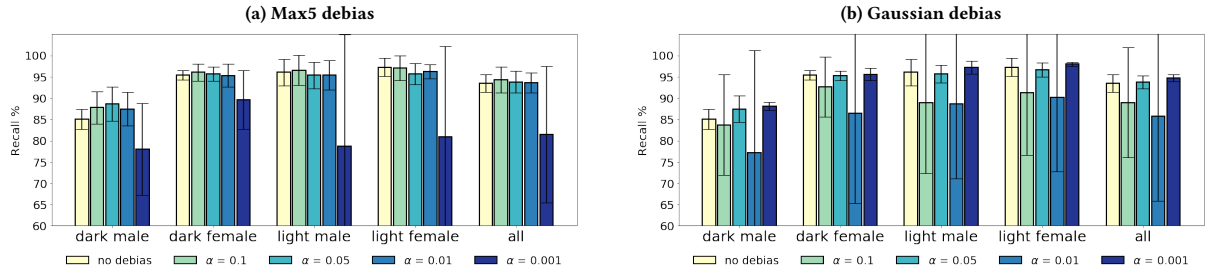


Figure 4: Bar plot of the recall of the different models trained using different degrees of debiasing. Plot (a) shows the results from the models trained using the max5 debias method. plot (b) shows the results from the models trained using the gaussian method.

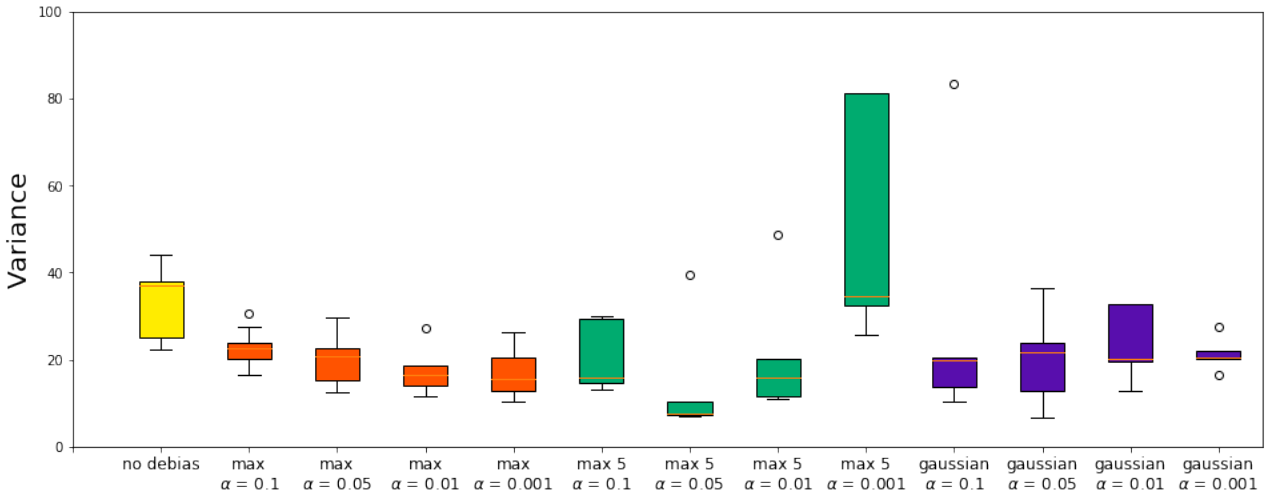


Figure 5: Boxplot of the variance between classes for all trained models. Each model is trained for 50 epochs.

Figure 6: Example of how subimages are created for evaluation.

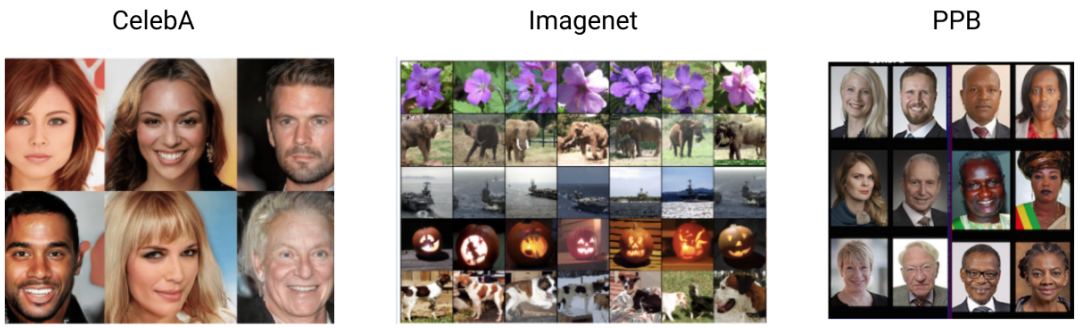


Figure 7: The datasets used for training (CelebA and Imagenet) and for evaluation (PPB).

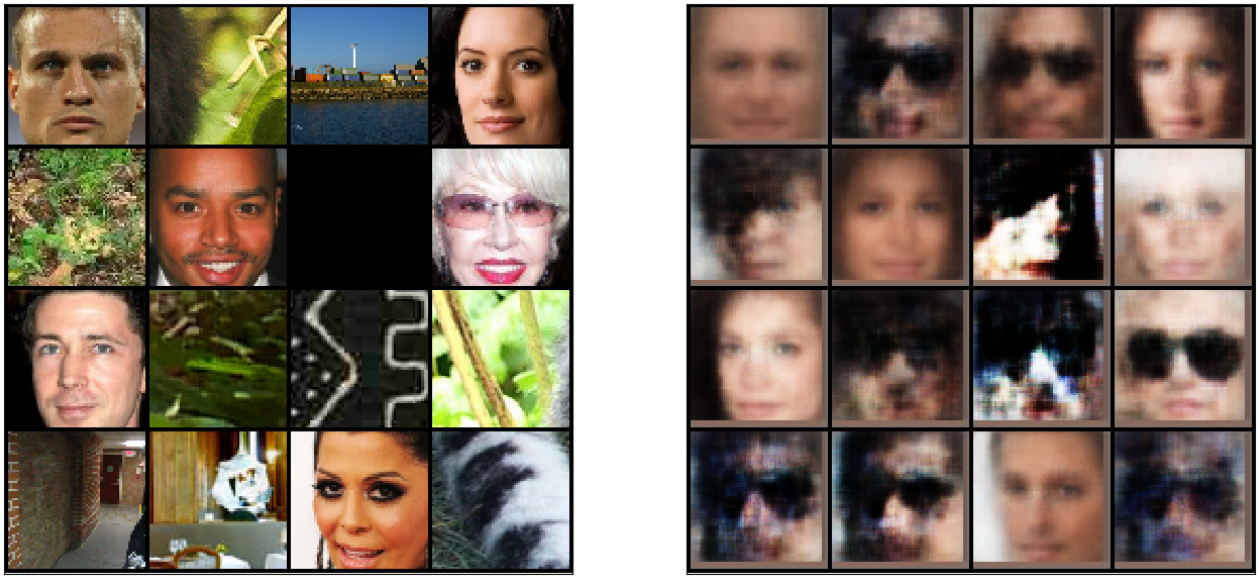


Figure 8: Examples of reconstructed images. The model only learns to reconstruct faces which is why non-faces are reconstructed very badly

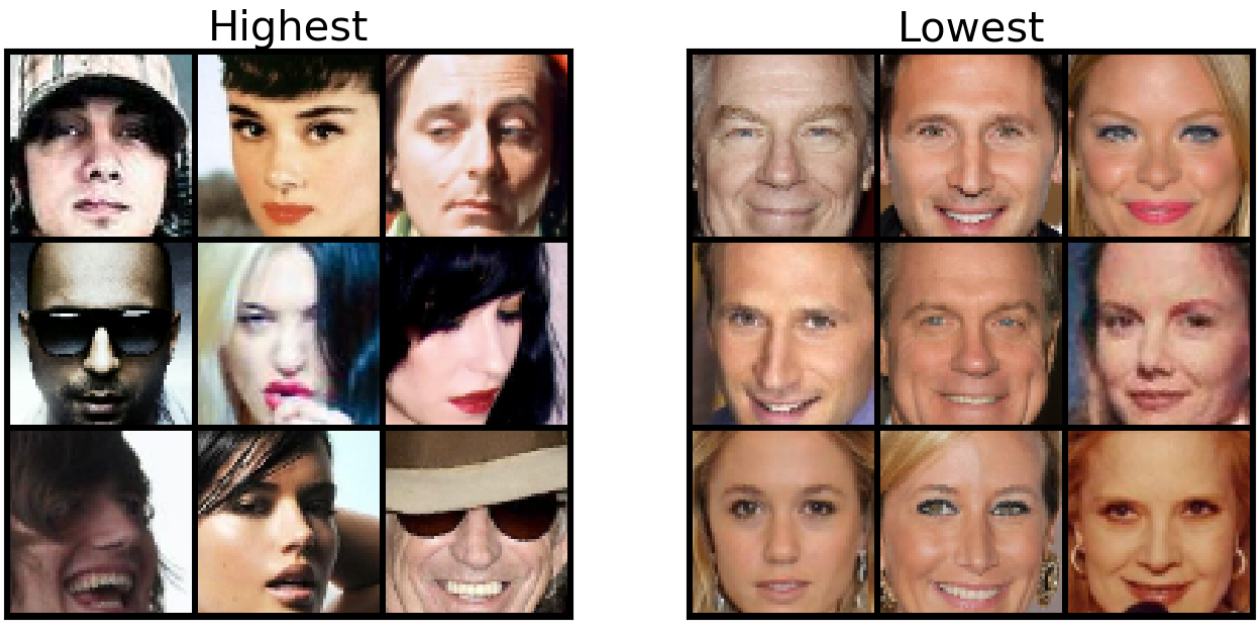


Figure 9: Examples of which images have the highest probability of being sampled and which have the lowest. The models rewards images that are less common like faces with hats with high probability, while common images are rewarded with a low probability.