Fair Face Recognition

Bence Zoltan Balazs (beba@itu.dk), Thomas Fosdam Claudinger (thcl@itu.dk)
IT University of Copenhagen, May 19, 2023

1 Introduction

The facial recognition technology has been proven to be problematic several times before, mostly because of racial and gender biases that are apparent in most algorithms and in photoagrphy in general [1]. We audit a popular facial recognition Python library and intend to explain which facial features help the algorithm detect a face on an image and report what kind of biases it might have.

1.1 The algorithm

In our project, we are auditing the face_recognition Python module from Adam Geitgey. The goal of the library is to "Recognize and manipulate faces from Python or from the command line"[2]. This includes recognising individuals based on their facial structure, and highlighting landmark features on their face such as mouth, eyes, nose etc. In our audit, we use the library's feature to find all faces on a given image. The underlying model of the face_recognition module is pre-trained for face recognition tasks. The model is a modified version of ResNet-34 from the paper "Deep Residual Learning for Image Recognition" by He, Zhang, Ren, and Sun[3] reduced to 29 convolutional layers. The authors claim "the model was trained on a dataset of approximately 3 million faces, derived from several sources: the face scrub dataset[4], the VGG dataset[5], and a large number of images scraped from the internet. They claim that "the model achieves a mean error of 99.38% with a standard deviation of 0.27% on the Labeled Faces in the Wild[6] benchmark"[7]. We do not have access to all the training images, thus we cannot determine how biased it might be towards different races, genders or age groups. However, we know that the distribution of identities in the VGG-Face dataset may not be representative of the global human population.[5]

1.2 Dataset characteristics

We evaluate the algorithm on the CelebA dataset. It was created by the Chinese University of Hong Kong to be used for attribute prediction in the paper "Deep Learning Face Attributes in the Wild"[8]. They used the already existing dataset Celeb-Faces and commissioned a professional labeling company to annotate each of the images in the original dataset using 40 binary labels (e.g. "pointy nose", "male", "smiling", "wearing makeup") and 5 key points. In total, the dataset contains over 200 thousand images and the labeling by 40 attributes resulted in over 8 million binary labels. The dataset was published together with their research paper on attribute prediction.[8]

From the dataset we are using two parts; images and a table of attributes (stored as a csv-file). It has 202,599 images and which are cropped to 178x218 pixels. The attributes table has the image filename and the 40 binary attributes for each image, 1 representing a feature being present and -1 representing a facial feature not being present [9]. There are no missing values in the attributes table, so no further cleaning is needed.

When looking at the distribution of attributes among the 200 thousand images, some of which can be seen in Figure 1. There are several underrepresented groups. For example, there are 60% females and only 40% males in the dataset. There are also very few that are chubby or have pale skin. Furthermore, there are more young people on the images than old ones. However, it is not clear from the original paper who they considered young during annotation.

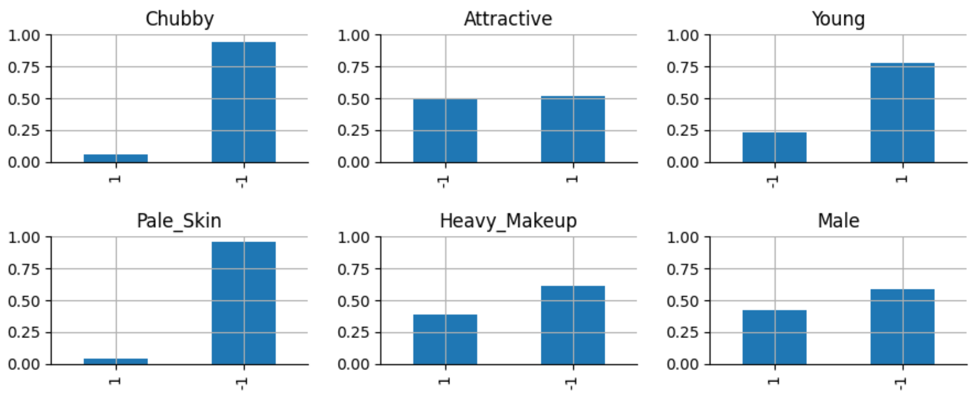


Figure 1: Distribution of labels for some attributes in the dataset. 1 being positive and -1 being negative.

1.2.1 Down sampling

After the first round of predictions we down sampled the dataset due to computation time in further experiments. We kept all the 5,609 images where the algorithm couldn't recognise any faces and randomly sampled 40,000 other images from the rest. After we compared the distributions we found that the feature distribution of this 40 thousand sample was almost identical to the approx. 195 thousand images that the face recognition algorithm could detect.

2 Methodologies

2.1 Image augmentations

We used image augmentations to study which features of the image that are not in the 40-feature binary set might influence the model, and whether they help to detect faces with some features that it could not otherwise. We used 4 kinds of augmentations. Firstly, we used a dark skin color layer on top of the images with the RGB code (1, 7, 38) with 60% transparency. Secondly, we used the same approach and added a light skin color layer with the RGB code (250, 231, 218) on top of the images again with 60% transparency. We intended to emulate the difference between groups of people with different skin tones. Based on our observations, people of European descent were overrepresented in the data set. As we can see in Figure 2, adding transparent layers on top of the images seemed to make the image more faded and therefore more difficult to detect faces. So, for the third augmentation instead of adding layers on top, we increased the brightness of the images by 50%. Finally, we increased the contrast on the images by 20%.

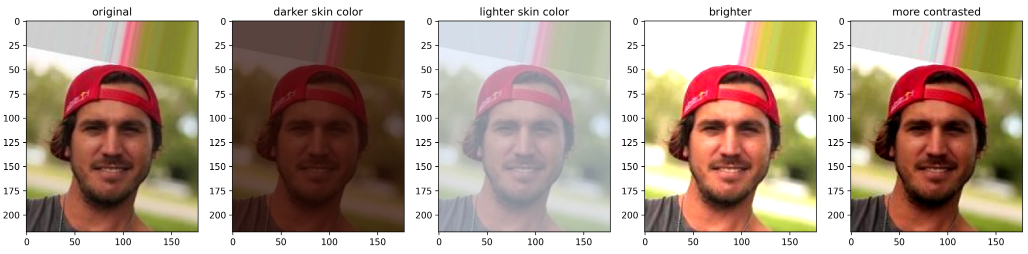


Figure 2: This figure shows the different augmentations of the images that we have added.

2.2 Covering areas of the image

We weren't able to access the weights of the dblib convolutional neural network which was the underlying model of the face_recognition module. Thus, we couldn't use libraries such as Captum or SHAP which could have shown which regions on the image had the most impact on detecting a face. Hence, we came up with our method where we covered a 40x40 pixel area in a sliding window with a stride of 3 pixels. At each step, we tried to detect a face in the image. For each of the pixels, we counted how many times it was covered and we could still detect the face in the image, then we normalised this count by the number of times when the pixel had been covered.

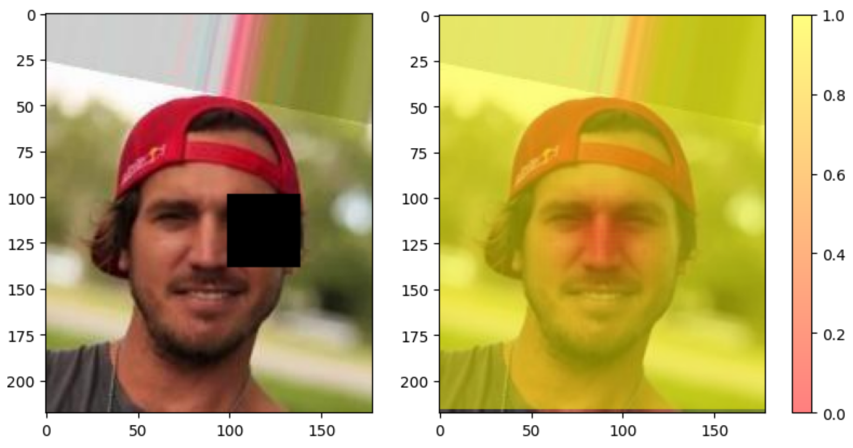


Figure 3: Example partially covered face on the right and a results heat map on the left

2.3 Decision tree

We wanted to use a white box model to see if we could identify any features that were related to whether or not the model detected faces. The idea was to train the model on the binary features with the target being either 1 if it detected a face or 0 if it didn't. The assumption here was that even though the features weren't used directly to detect faces, they could be seen as latent variables. We chose a decision tree as it gives importance to the features while being easy to interpret afterward.

Before training the decision tree we balanced the labels in the training set since a skewed distribution of labels would probably skew the outcome as well. From the balanced training dataset we did a grid search and cross-validation, using the GridSearchCV method provided by Scikit-Learn [10], to find the best parameters. Finally, we made predictions on the unbalanced test data and also looked the how the classifier scored the features by importance.

2.4 Statistical Parity

We wanted to measure how well the algorithm detected faces in relation to the attributes. Since we have a dataset with faces on all images we can measure True Positives, TP (faces detected that are actually there) and False Negatives, FN (faces not detected that were actually there). This makes us able to calculate the Statistical Parity for the protected features. Looking at the non-protected features the True Positive Rate is calculated the same way as statistical parity, as shown below.

$$\text{Statistical Parity} = \frac{\text{TP} + \cancel{\text{FP}}}{\text{TP} + \text{FN} + \cancel{\text{FP}} + \cancel{\text{TN}}} = \frac{\text{right guesses}}{\text{right guesses} + \text{wrong guesses}} = \text{Accuracy}$$

We don't have any true negatives or false positives, so we aren't able to calculate fairness metrics such as Equalized Odds etc.

Based on our preliminary data exploration we decided to use gender, attractiveness, skin colour, and age as protected features. We used gender, skin colour and age because previous research has shown that models can be biased toward white middle-aged men. We added attractiveness because, during the first run of the model, we found that it detected faces more often that were blabeled as attractive, and in our opinion, such a relative measure should not influence a face detection algorithm.

2.5 Model accuracy

As mentioned in the previous section we can easily calculate the True Positive Rate. In the case of not having images without faces, the True Positive Rate tells us the accuracy of the face_detection module on our dataset.

3 Results

There are 39,327 images where the model could detect a face with all augmentations in every case and 4,869 where the model could never detect a face. We compare some of the feature distributions of these two groups in Figure 4 and we will compare the remaining 1,413 images and analyse how an augmentation helped to recognise certain features in Figure 5.

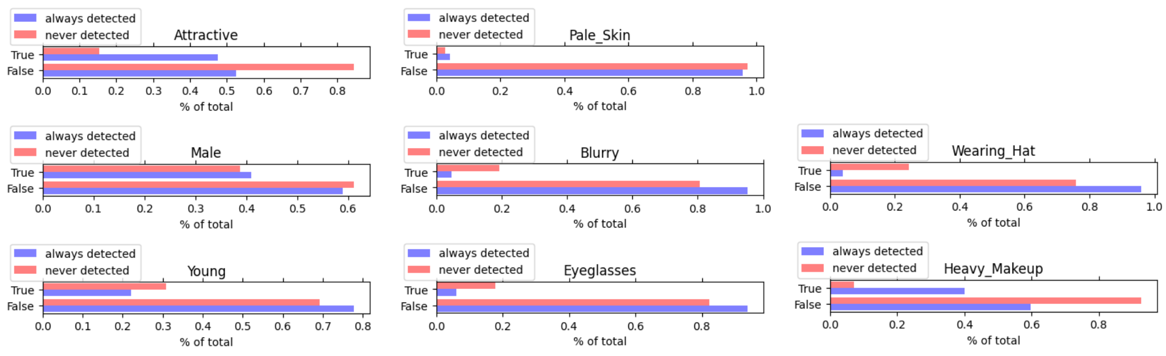


Figure 4: This figure shows the distribution of all features on the images where the model could either always or never detect a face

We found that features that indicate that some of the face is covered, for example, when the person in the photo is wearing a hat, or eyeglasses or the image is blurry, made it more likely for the model to fail to detect the face. Surprisingly, if an individual wore heavy makeup it helped the model to detect a face, likely because it emphasizes certain facial features that the model is looking for. Other features, like pale skin and gender, were relatively balanced between being recognized or not. Being young also made a face less likely to be detected by the model.

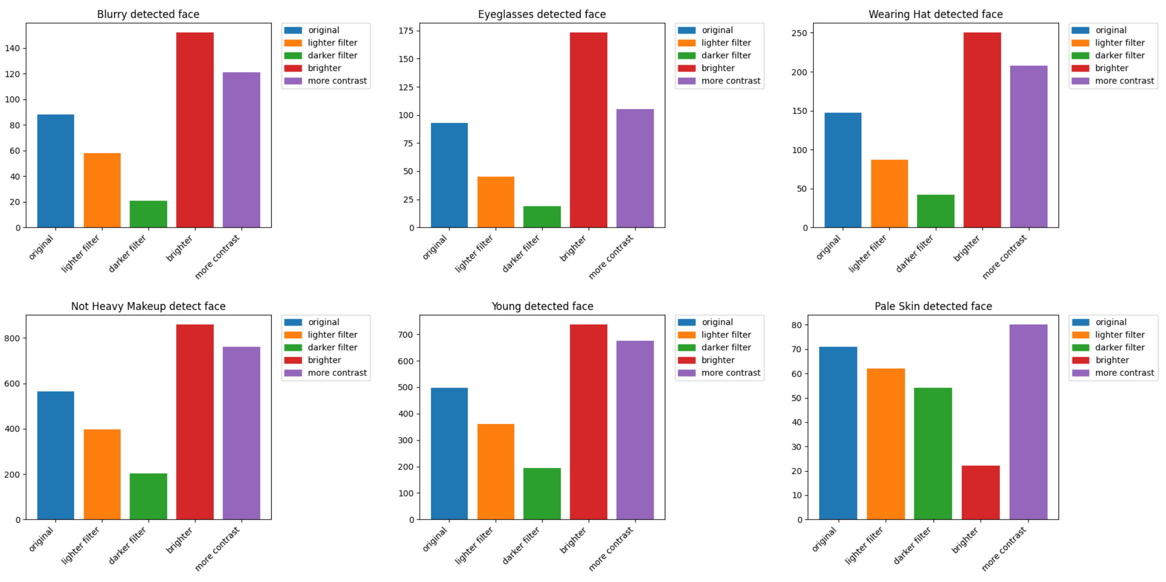


Figure 5: These plots show that on how many images could the model detect a face with a given feature that it could not detect in all the cases.

We can see in Figure 5 that in most cases making the images 50% brighter can influence the model the most to detect of the images that it could not detect before. However, in the case when the person in the image has pale skin it underperforms. Also, we need to note that fewer individuals in the images had pale skin compared to other attributes. When we increased the contrast on the images by 20%, the model consistently achieved better results than on the images without any augmentation, even in the case when people had pale skin. The algorithm had worse results persistently on the images with the two skin tone filters, compared to the unmodified images. However, it is worth mentioning that all augmentation was able to help the model on images could not detect faces on without augmentations (see Figure 7 in Appendix B).

3.1 Focus areas on the faces

Running our focus area detection algorithm is very time-consuming. It takes about 30 seconds for a single image. Thus, we only did qualitative analysis on a random sub-sample of images.

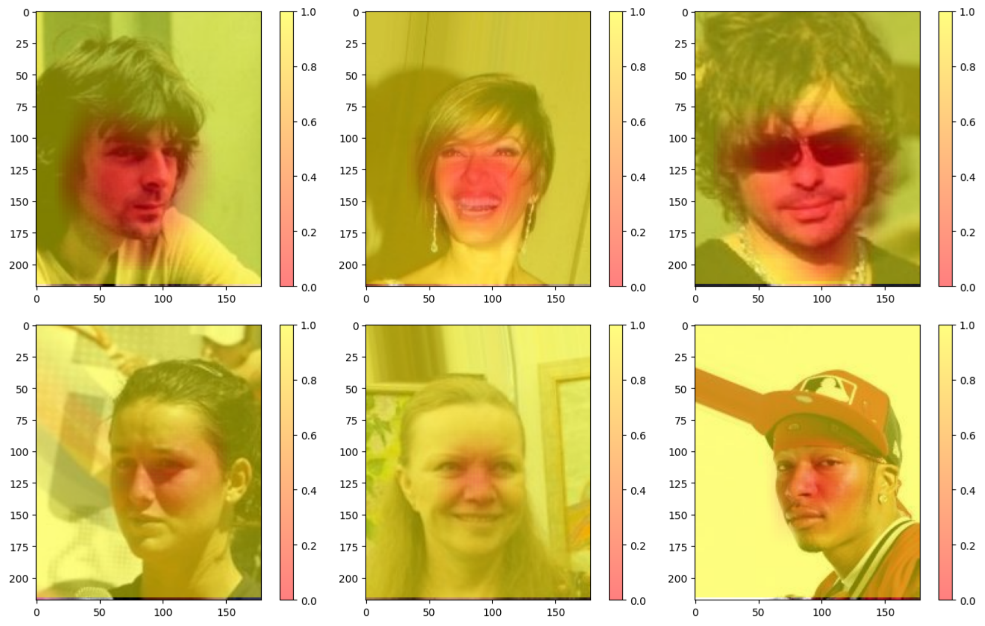


Figure 6: This pictures show the ration when a pixel has been covered and the face still has been detected 1 meaning every time 0 meaning never

We found that if parts of the face were covered by hair, sunglasses, or masks the algorithm became more sensitive to covering other parts of the face. Also, when the image was taken from the side the algorithm was sensitive to a slightly larger area. In most cases, when an image was taken from the front the algorithm paid more attention to the area around the nose, between the eyes, or over the lips.

3.2 Decision tree

As mentioned in the methodology section 2.3, we split the downsampled normal (not augmented) data into train and test. We balanced the training set and trained a decision tree classifier using parameters found with GridSearchCV. The grid search resulted in the following parameters: *{'max_depth': 12,* *'max_features': None, 'min_samples_leaf': 5, 'min_samples_split': 15}*. A classifier with these parameters gave an accuracy score of 0.77 and a F1 score of 0.85 on test data.

The white box model could predict with this accuracy and precision whether the face_recognition module would detect a face on an image based on these 40 binary features. This seemed to be a fairly good model and was a bit in our favor of suggesting that attributes could be used as latent variables. From the feature importance score of the decision tree, we could see which attributes it put more weight on when predicting as shown in Table 1. The full list can be found in Appendix A.

Table 1: Feature important scores, five highest and lowest scored attributes.

Feature	Importance
High_Cheekbones	0.372
Attractive	0.182
Arched_Eyebrows	0.079
Wearing_Hat	0.072
Bags_Under_Eyes	0.043
...	...
Pale_Skin	0.000
Goatee	0.000
Sideburns	0.000
Mustache	0.000
Bald	0.000

3.3 Statistical Parity

We looked at all possible combinations of our 4 protected features which were binary – in total 16 combinations. For each combination, we calculated Statistical Parity. This was repeated for the normal dataset and the 4 augmented datasets in order to find out whether we managed to change some of the statistical parity by augmentation. The results in Figure 2 show that the *young, unattractive male without pale skin* had the lowest parity for the normal dataset, and the augmentation with increased brightness increased the statistical parity.

Table 2: Statistical Parity for all combinations of protected features and augmented datasets.

skin	gender	age	attrativeness	% people	normal	lighter_skin	darker_skin	brighter	contrast
pale skin	male	young	attractive	0.50	0.948	0.952	0.943000	0.930	0.952
pale skin	male	young	not attractive	0.27	0.815	0.815	0.815000	0.798	0.823
pale skin	male	old	attractive	0.02	0.909	0.909	0.909000	0.909	0.909
pale skin	male	old	not attractive	0.21	0.844	0.823	0.823000	0.802	0.854
pale skin	female	young	attractive	2.26	0.952	0.948	0.948000	0.928	0.954
pale skin	female	young	not attractive	0.53	0.820	0.803	0.783000	0.775	0.824
pale skin	female	old	attractive	0.14	0.985	0.985	0.985000	0.939	0.985
pale skin	female	old	not attractive	0.21	0.885	0.885	0.875000	0.875	0.906
not pale skin	male	young	attractive	9.35	0.959	0.956	0.954000	0.963	0.962
not pale skin	male	young	not attractive	17.24	0.775	0.769	0.761000	0.790	0.783
not pale skin	male	old	attractive	0.96	0.959	0.957	0.947000	0.954	0.961
not pale skin	male	old	not attractive	14.96	0.800	0.793	0.787000	0.809	0.804
not pale skin	female	young	attractive	32.48	0.957	0.955	0.953000	0.959	0.959
not pale skin	female	young	not attractive	14.22	0.811	0.804	0.796000	0.827	0.819
not pale skin	female	old	attractive	2.00	0.982	0.980	0.978000	0.979	0.984
not pale skin	female	old	not attractive	4.65	0.886	0.883	0.879000	0.887	0.890
mean					0.893	0.889	0.884	0.883	0.898
standard deviation					0.004949	0.005433	0.005746	0.004829	0.004649

Looking at Table 2 on the *% people* column, we notice that the distribution of people in each protected groups is skewed. If we look at the statistical parity, specifically individuals not having pale skin, and zoom in on younger individuals in this group, we see the statistical parity increased from the normal to the brighter augmentation. For individuals that were old, it decreased a tiny bit or almost stayed the same. The observation gives a small indication that face detection is easier on individuals with lighter skin, and increasing the brightness unbiases the model a bit.

If we now look at individuals with pale skin and compare normal with contrast augmentation, we see that the statistical parity for all individuals with pale skin increased when increasing the contrast. This could indicate that increasing contrast in general makes face detection easier for the model.

For pale and not-pale individuals overall the mean of the statistical parity increased and the standard deviation of statistical parity decreased on the brighter dataset compared to the normal. Again, looking at the contrast dataset we see the same, increased mean and reduced standard deviation. The contrast augmentation received the best score out of all 5 datasets.

3.4 Model accuracy

We used the True Positive Rate to report on the model's accuracy. We found that three features have significantly lower accuracy than the rest of the features; *"Blurry, True"*, *"Eyeglasses, True"* and *"Wearing_Hat, True"*. These correspond to the ones that are obvious if we think about what would make it difficult to detect a face in an image. We compared the overall accuracy of the model on the original dataset and the four augmented datasets:

Table 3: Accuracy score on normal and augmented datasets

normal	darker_skin	lighter_skin	brighter	contrast
0.882	0.873	0.878	**0.888**	0.887

We can see in Table 3 that the model's accuracy score is better only by 0.001 on brighter images compared to contrast increased images. In general, the model performed better on both brighter and contrast than normal. However, on the darker_skin and lighter_skin dataset it performed worse.

4 Discussion

Our results show that the most important of the features that we have chosen as protected is the attractiveness of the individual in the image. Although, we need to note that what is considered attractive is highly cultural and, in our dataset, it shows a high correlation with features such as wearing heavy makeup or wearing lipsticks. Further characteristics that we can read from the result of the decision tree feature importances is that features that are strong facial characteristics such as high cheekbones, arched eyebrows, bags under eyes also highly influenced the model. Also, as mentioned before when a part of the face is covered by a hat or the image is blurry it often makes the model less accurate.

The evaluation of statistical parity for protected features - pale skin, gender, age, and attractiveness - revealed that the model performed marginally better on a dataset augmented with brightness and contrast, compared to the original dataset. More specifically, the brightness filter appeared to increase the statistical parity for young individuals with non-pale skin. These findings suggest that the model may exhibit some bias toward this particular group. However, the model gave the highest average statistical parity on the contrast data. The True Positive Rate used to report the model's accuracy showed the same tendency – that the model performed best on the brighter and contrast.

Overall we could make the algorithm more balanced for all the protected subclasses by increasing the contrast of the images. This might be because it makes the facial features more dominant in the images. Increasing image brightness also increased the performance of the algorithm. However, as these two augmentations are specific to this dataset we cannot confidently state that increasing the brightness or contrast of the images will help the algorithm to be fair. We also examined that overlaying a skin colour filter on the images doesn't increase the performance of the algorithm, but we can still observe that applying the darker skin tone filter decreased the performance of the model more than when we applied a light skin color filter on the images.

5 Ethical Discussion

The question of whether we should use face recognition algorithms is as old as the facial detection field itself. There are concerns about individuals' privacy or the systematic discrimination of groups of people. On the other hand, there are the uses that make our everyday lives easier, for example, auto-focusing on our phones when we take an image or using face recognition as an authentication method on our phones.

If we take the Kantian ethics perspective, the people should not be used to achieve a goal but the goal should be for people to have better lives. We should also consider privacy a fundamental right. Thus, it is crucial that whenever a facial recognition/detection algorithm is used on a person they can make a well-informed, active, voluntary decision. As most facial detection models are black box models, it is crucial that we explain them properly and in-depth to everyone before we use it on them. We could do this by doing similar analyses as we did in this report, phrasing the results in a simple and clear manner and showing examples of where the face algorithm is focusing and informing them about the privacy and security concerns that appear whenever a similar algorithm to the face_recognition Python library is used. This entails that these algorithms should not be used on CCTV cameras in public spaces or in any in the judicial system but could be used on phones for authentications or when taking selfies.

On the other hand, if take a utilitarian point of view we need to focus on maximising overall happiness and well-being. This entails that we can use these algorithms to enhance public safety, prevent potential threats and and streamline identification processes. This could reduce crime rates and help in situations when an identity check is required, like passing through a border. It could also be used to personalise user experiences in the online space with tailored services and personalised ads that are relevant to the user. However, with this perspective in mind, we would constantly need to monitor if the users of the technology are content with the results and if it is still beneficial for most of the stakeholders and not discriminating against any groups.

It is also worth considering why we are choosing to do face recognition/detection using such a complex black box model. Why can't we use a model that can be easily interpreted by anyone, like a decision tree? To answer this we need to look into the theory of how we interpret faces. We have a global understanding of a face, we are able to identify parts of it like the nose, ears, eyes etc. We combine these with many abstract representations in our minds, like emotions and expressions. Moreover, we are able to update these representations constantly by thinking about them, "doing operations on them". Meanwhile, simple algorithms that are easy to interpret are unable to have this kind of complex representation of a face. Thus, we need to use more complex and general estimator functions, a neural network. In our case, a residual neural network which creates a high dimensional vector that represents features of the face, although these might not correspond to any of the parts that we conceptualize as humans. Also, this representation is highly dependent on the training data; if there are parts of the face that it has not seen, it might simply be unable to detect a face. Many times adding random noise can completely confuse these algorithms because it has never seen it before.

6 Limitations and Future works

One of the limitations of this project is that we only used one dataset. In this, all the images contained a face, this way we couldn't test if the algorithm recognises objects that only look similar to a face but aren't one, like an artificially generated face. We could have used an artificially generated face dataset. In future, we could use further fairness metrics this way, like equalized odds, or equal opportunity for different groups, as we would have false positive and true negative results. At the beginning of our project we also falsely interpreted the attribute pale as referring to "white" people, however, it refers to very light skin color. In future experiments, we would not treat that feature as a protected group. Also it made our protected subgroups unbalaced as there were significantly less people with pale skin as seen in Table2.

We would also like to be able to make the model easier to interpret in the future by using libraries such as Captum or Shap. To do this, first we either need to be able to load the network with its weights into Pytorch or TensorFlow, or we would need to try to train another algorithm with the same architecture. This way we could de-bias the algorithm by training it on a fair and balanced training set and explain focus areas based on network weights.

References

[1] Sarah Lewis. The racial bias built into photography. *The New York Times*, April 2019. URL https://www.nytimes.com/2019/04/25/lens/sarah-lewis-racial-bias-photography.html.

[2] Adam Geitgey. face_recognition: A python library for face recognition. https://github.com/ageitgey/face_recognition, 2023.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[4] Theo Things. facescrub-dataset, 2018. URL https://github.com/theothings/facescrub-dataset.

[5] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 2018. URL https://arxiv.org/abs/1710.08092.

[6] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49, 2007. URL http://vis-www.cs.umass.edu/lfw/.

[7] Davis King. dlib-models, 2017. URL https://github.com/davisking/dlib-models.

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[9] Jessica Li. Celebfaces attributes (celeba) dataset, Jun 2018. URL https://www.kaggle.com/datasets/jessicali9530/celeba-dataset.

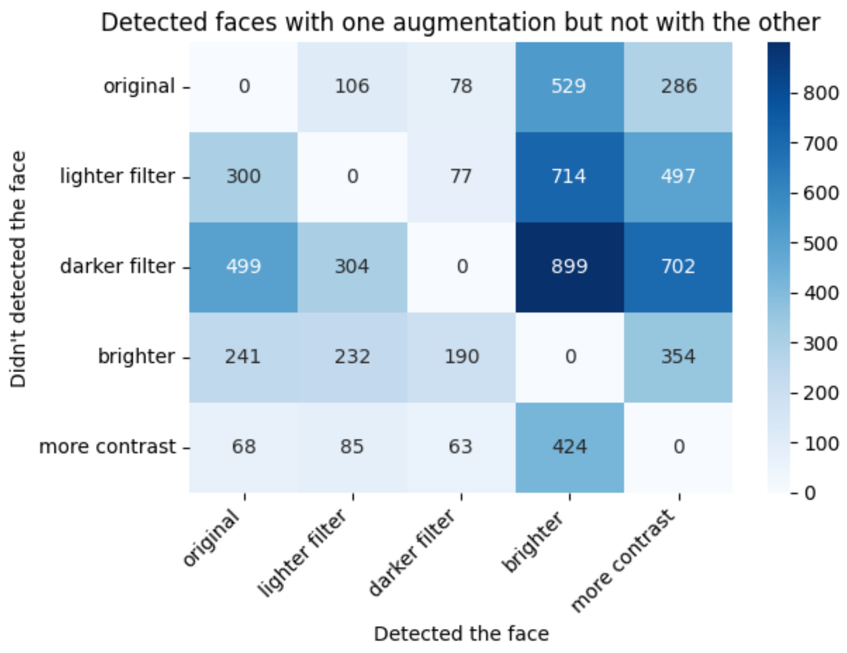
[10] Scikit-learn documentation. URL https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

A Feature importance from DT

List of the decision tree classifiers score of feature importance for all attributes, sorted in descending order of importance.

Feature	Importance
High_Cheekbones	0.372
Attractive	0.182
Arched_Eyebrows	0.079
Wearing_Hat	0.072
Bags_Under_Eyes	0.043
Oval_Face	0.034
5_o_Clock_Shadow	0.033
Wearing_Lipstick	0.026
Smiling	0.020
Heavy_Makeup	0.015
Blurry	0.014
Straight_Hair	0.013
Bushy_Eyebrows	0.013
Bangs	0.012
Male	0.008
Wearing_Necktie	0.007
Chubby	0.006
Big_Nose	0.006
Brown_Hair	0.005
Black_Hair	0.004
Young	0.004
Double_Chin	0.004
Mouth_Slightly_Open	0.004
Pointy_Nose	0.004
Receding_Hairline	0.003
Wavy_Hair	0.003
Wearing_Earrings	0.002
Blond_Hair	0.002
Narrow_Eyes	0.002
Eyeglasses	0.002
Rosy_Cheeks	0.001
Gray_Hair	0.001
Big_Lips	0.001
No_Beard	0.001
Wearing_Necklace	0.001
Pale_Skin	0.000
Goatee	0.000
Sideburns	0.000
Mustache	0.000
Bald	0.000

B Augmentations



	original	lighter filter	darker filter	brighter	more contrast
original	0	106	78	529	286
lighter filter	300	0	77	714	497
darker filter	499	304	0	899	702
brighter	241	232	190	0	354
more contrast	68	85	63	424	0

Figure 7: Shows in the columns with which augmentations could the model detect a face while it couldn't with the other