**Verifiable Fairness in Biometrics**

**Motivation**

The underlying mechanisms behind biometric authentication and facial verification systems are often prone to demographic bias [1, 2], due to unbalanced representations, labelling bias, algorithmic bias or hardware limitations. A lack of consideration for bias when developing facial recognition and detection models and infringe the rights of individuals and lead to structural racism and systematic marginalisation [3].

**Introduction**

MOSIP and OpenG2P are open-source platforms which develop digital identity services (DIS). An example is the OpenG2P 4Sure verifier app[[1]](#footnote-1) used for offline verification of citizen identity, which makes use of biometric authentication systems. These biometric authentication systems are developed by third-party vendors, i.e. the underlying machine learning models and their development pipelines, including data curation/generation, and model development and testing. In the case of both MOSIP and OpenG2P, these systems are then integrated into the digital identity service via the Biometric SDK[[2]](#footnote-2), a third-party tool for integrating biometric systems.

Biometric systems are typically tested using metrics such as FPRs, FNRs, accuracy etc [4], and the system integrator (such as a government) would set acceptable thresholds for these metrics. However as mentioned previously, these systems are prone to demographic bias and thus it is in the best interest of MOSIP or OpenG2P to have assurance that the underlying mechanisms used in their services are fair to individuals, despite being agnostic to the model development process[[3]](#footnote-3),[[4]](#footnote-4). As there is no standardised fairness testing within biometrics [5], along with some authors suggesting biometric models are intrinsically biased despite mitigation methods [6], it is important to provide transparency regarding the existing bias of biometric systems used by and on individuals. This can be done once the model is trained and will be discussed later, which is advantageous when the data used to train the model is unavailable. In the case where the training data is available, assessing the bias in the data processing stage of model development can be paired with existing methods in verification of datasets, enabling the proof of fairness for the data used to train the model [7].

**OpenG2P use case**

OpenG2P provides the 4Sure verifier app[[5]](#footnote-5) which is used for verifying digital identities offline. Within the app, there is the Face Match SDK module, which is used for facial verification, i.e. a biometric model would be used to match the user provided image (of themselves) with an existing image in a database (such as a national ID). An organisation or government making use of the 4Sure verifier app would need to use a vendor to develop the underlying biometric authentication mechanism to be used with the 4Sure verifier app, via the Biometric SDK.

A diagram of a model

Description automatically generated

**Figure 1** - Overview of the pipeline for facial verification used by the 4Sure verifier app. The vendor produces the model (data collection, processing, model training) for the facial verification task which is then implemented into the 4Sure app via the Biometric SDK. OpenG2P is agnostic of the model training pipeline and testing.

**MOSIP use case**

MOISP uses biometrics for verification of digital identity[[6]](#footnote-6). Similar to the OpenG2P use case, a vendor develops the underlying biometric systems which are then implemented into the MOSIP tool via the same Biometric SDK as with OpenG2P. The biometric system is used for 1:1 authentication to match an individual to biometric details (fingerprints, iris and face) present in a MOSIP ID Repository[[7]](#footnote-7). The Biometric SDK is implemented into other various points in the pipeline for different purposes, including quality check (which checks the quality of the input biometrics). In addition to 1:1 authentication, MOSIP utilises Automated Biometric Identification Systems (ABIS) for 1:N deduplication. Deduplication compares new biometric data submitted via the registration process against existing data in the database. If a match is found, then the registration fails and is investigated. The ABIS is a separate process to the Biometric SDK, and instead interacts with MOSIP via only message queues. These are used due to the matching process having to cross-reference through every record in the gallery along with extra security measures being in place. While different processes, if the underlying models are developed similarly to the ones used in 1:1 authentication, then the same risk applies due to the inherent bias of such models, at least in the case of facial verification.

A diagram of a model

Description automatically generated

**Figure 2** - Overview of the pipeline for MOSIP biometric authentication using the biometric SDK. Similar to Figure 1 with the OpenG2P use case, the vendor develops the model. The model endpoint interacts with the Biometric SDK integration points within the pipeline, for different tasks, including authentication, deduplication and quality checking.

A biometric model can discriminate against particular demographics as previously mentioned, or against other sensitive attributes present in the individuals they are used on or with. This affects individuals using the digital identity services, as for example, a service used for facial verification may have a better accuracy for individuals from one ethnic group over another. This would lead to some individuals not having their identity verified correctly.

**Verifiable Fairness**

Various methods exist to test and mitigate the fairness of biometric systems [8-10]. In theory anyone with access to the weights of a model can test the fairness of a model. The issue with this is firstly, not all models or the datasets they are trained on are open access. Secondly, the integrity of the assessment reduces to the trustworthiness of the entity carrying out the assessment, and likewise there would be no way to prove the test result actually belongs to the specific model in question. Thirdly, these audits may require the use of reference datasets which themselves may be biased. Thus, we require a method for demonstrating the fairness of a model, without revealing the model weights, in such a way that is verifiable and associative with the model.

The term “verifiable” in this context refers to some metric or measure, such as individual fairness, being secure (i.e. tamper proof) and provable to a user, without the need to expose the model weights or data it was trained on to such user. Individual fairness is an important fairness notion in this application as we require biometric systems to be individually fair, i.e. the output of the model should be the same for different images the same individual. For two inputs of the same individual, , a model is individually fair if . A parameter can be used as a measure of individual fairness when it measures how much a feature can change before the output of the model changes, for an input pair. For example, how bright or dark an image can be of the same person, before the model’s output changes.

The authors of “*FairProof: Confidential and Certifiable Fairness for Neural Networks*” [11] use the parameter as a fairness certificate, which a user (such as the vendors creating biometric models or even MOSIP/OpenG2P, if they have access to the model weights) can then verify whether the fairness is computed correctly or not, without revealing the model weights to any interested external party, such as an auditor or the end user such as the individual using the digital identity service. This is done using Zero Knowledge Proofs (ZKPs) [12]. It is key to note that the methodology of FairProof is to evaluate the fairness of a model at a specific data point, rather than for the entire input space. In other words, it verifies the fairness of the model for a particular individual, as opposed to the fairness of the model over a database which is sufficiently representative of the demographics the model will be used with.

A diagram of a graph

Description automatically generated

**Figure 3** – A fully connected feed-forward neural network with ReLU activation functions is a collection of piecewise linear functions. It forms a decision boundary highlighted by the orange line. For an input x, the decision of the model (e.g. whether a face matches or not) changes depending on the features (which may be sensitive) of x. measures how much this feature can change before the input x moves over the decision boundary, i.e. the models output changes due to the change in the (sensitive) feature.

**Proposal**

For a model used in facial verification, the development pipeline, broadly speaking consists of firstly the curation or generation of a dataset containing images of individuals. There are numerous instances of each individual, and the model (usually a CNN) is trained to match input pairs of individuals, with the result at the inference stage being a binary classification, i.e. “true” or “false” to whether an input pair represents the face of the same person. A fairness assessment can take place at the inference stage of this process once the model is trained, and/or during the training process itself if the training data is available.

There are four different approaches that can be taken to improve the transparency of the pipelines used in the production and usage of biometrics in digital identity services hosted by MOSIP or OpenG2P. Each approach seeks to remove the trust related issues [13, 14] with canonical auditing methods by using verifiable credentials.

1. Attest to a models fairness by demonstrating the fairness of the data used to train it, using data provenance techniques, found in work such as “*Certifying provenance of scientific datasets with self-sovereign identity and verifiable credentials.*” [7]
2. Demonstrate a model’s group fairness a fairness testing dataset, aiming to test fairness for specific features such as skin colour, without the need for the original training data.
3. Demonstrate a model is individually fair for particular individuals, providing certificates for each individual, as suggested in FairProof, again, without the need of the original training data.
4. A mixture of the above.

A direct use case of this work within MOSIP is through the MOSIP Compliance Toolkit (CTK)[[8]](#footnote-8) which is an online portal used by MOSIP partners (e.g. vendors developing biometric systems) to ensure the solution is compliant with MOSIP specifications. An example of a compliance test for ABIS systems is defined in the ABIS API specifications[[9]](#footnote-9), of which, defines target false-positive identification rates (FPIRs)[[10]](#footnote-10) among other operations. Currently there are no compliance tests for measuring bias within biometric models. The result of this work can be used to develop a MOSIP CTK with to-be-determined target thresholds such as equalised odds, for biometric models used in MOSIP and OpenG2P solutions.

**Next steps**

The methodology will evolve from the following:

1. Selection of a biometric standard
2. Selection of sensitive features to test for bias against, fairness notions and metrics. Then curation or generation of a dataset
3. Selection of a model
4. Evaluation of group fairness
5. Evaluation of individual fairness

For our experimentation, we can use the ICAO guidelines for passport photos which will be relevant to biometrics[[11]](#footnote-11). These guidelines outline image size, acceptable exposure levels, presence or absence of accessories among other salient points. We then select a sensitive feature which we want to avoid discriminating against, an obvious choice would be skin colour as models are prone to bias for this feature [15, 16]. In the context of facial verification, we are interested in the model correctly and consistently matching an individual to themselves. This can be measured by the True positive rate (TPR) where an individual is correctly matched to themselves, along with the False positive rate (FPR), where an individual is incorrectly matched to themselves, i.e. when the model is presented with two images of the same individual, it does not match them.

The Monk Skin Tone (MST) scale[[12]](#footnote-12) defines a 10-shade colour scale describing human skin colour, developed by Ellis Monk in collaboration with Google. One use case for these shades is the annotation of human skin colour within datasets, i.e. rather than sorting individuals by their race or ethnicity, sorting them by these skin colour tones. This is useful as some races or ethnicities have a wide range of skin colours within them. A testing dataset constructed of different individuals sorted into the 10 skin tone groups according to MST can be generated using a similar methodology to SynthPar2[[13]](#footnote-13), as shown in **Figure 4**. Similar datasets exist, including Microsoft DigiFace [17] and synthetically altered ethnicities [18] although we require a dataset that separates individuals according to some demographic such as skin colour. This proposed open-source fairness testing dataset will be used for benchmarking biometric models and the result of this testing will a) form a compliance test within the MOSIP CTK and b) form a verifiable credential allowing viewers to be certain the test is carried out using the model and datasets of interest.

A diagram of people with different colors

Description automatically generated with medium confidence

**Figure 4** - Structure of a dataset which can be used to certify individual fairness across an entire input space. For each demographic group (such skin colour), there are unique individuals. For each individual, there are variations of that individual. The variation could be a result of different lighting conditions, or the presence/absence of features such as accessories or hairstyles, etc. These variations need to be compliant to some specification, such as the ICAO specification for passport photos. There are multiple different demographic groups containing identically arranged individuals.

A collage of a person

Description automatically generated

**Figure 5** - Example images of a person in "Monk skin tone 9"[[14]](#footnote-14). A dataset of multiple instances of the above, i.e. an individual and various images of them, from all 10 groups, can be used to measure a models bias against skin colour. Note in our use case, the images cannot vary as much as above in terms of pose or lighting.

DeepFace[[15]](#footnote-15) is an open-source Python library allowing users to run inference on SOTA facial verification and detection models such as VGGFace, FaceNet, ArcFace and others, with little code. The codebase can also be edited to incorporate any other user provided model, such as one developed by a vendor for use with MOSIP and OpenG2P biometric tools. Using the testing dataset as described above, we can calculate, for a given model, the TPRs and FPRs for each individual, in each demographic group. This is then repeated over all demographic groups. The results of this are then reported to a JSON file. This JSON file along with the model weights and testing dataset can be appended as a verifiable credential, proving to a viewer that the results belong to this particular model and the testing was done using this particular dataset.

The considerations for the above experimentation would be firstly: the time and cost of generating a facial verification dataset. Secondly: the time and cost of running inference over all input pairs. For individuals (according to **Figure 4**) there are input pairs. So for there are 49 input pairs for each individual. Given inference times for verification typically take ~50-500ms depending on GPU resources, testing all input pairs in a dataset containing a sufficient number of total individuals (perhaps in the order of ) could take hours or even days. At this point we have a methodology for testing a model’s group fairness according to groups of individuals in different skin tone categories. This can be repeated for other sensitive features also.

To next certify individual fairness, we follow the methodology of FairProof and firstly consider the authors apply this to shallow networks with very small dimensionality (2 hidden layers), in the context of tabular data (e.g. the German Credit scoring dataset). Applying this to the use case of facial verification using CNNs is not trivial and the authors of this work state as such. We then need to investigate the feasibility of abstracting the decision boundaries of CNNs. In the case of simple fully connected networks with ReLU (such as the ones described in FairProof), the decision boundaries are characterised by hexagonal polytopes as shown in **Figure 3**, and methods exist for defining them [19]. For CNNs, these boundaries may be high dimensional manifolds (as there are more features used to make a decision) and there may not be existing implementations on how to abstract such boundaries. Whilst there are fully connected layers at the end of a CNN, they are much larger than the layers used in the implementation of FairProof, which may cause expensive or even infeasible computational overheads when generating the ZKP. The second consideration is the input point, in the case of FairProof, this is a vectorised row from tabular data, in this case, it would be a matrix containing pixel values which again may result in expensive/infeasible computation. A starting point could be to augment facial data into tabular data and use a linear classifier for verification reducing the model dimensions down significantly. The implementation of FairProof[[16]](#footnote-16) requires 1) the decision boundaries of the layers of the network, 2) the weights of the layers, 3) the input point as JSON files. If it is possible to abstract this information from such larger CNN models, then a direct implementation thereafter may be possible assuming it is computationally feasible, but we will begin by using smaller linear models.

**Timeline**

Generation of testing dataset: 1-2 weeks, SynthPar2 may already be usable. If a new dataset needs to be generated, this will take similar time to SynthPar2 although we require far less examples

Testing group fairness for SOTA models through DeepFace: 1 week to write code, actual testing time depends on compute resources, anticipating this will take in the order of hours to days

Appending group fairness results into VCs: 2-3 weeks

Developing linear models for facial verification for individual fairness: 3 weeks

Using FairProof methodology on the above linear models, for individual fairness certificates: 1 week

**Conclusion**

We aim to benchmark the group fairness of biometric models used in MOSIP and OpenG2P services by using an open-source testing dataset (by curation or generation) according to different demographics, such as skin colour. The dataset and methodology of this testing is used to form a MOSIP CTK (Compliance Tool Kit). We then aim to test a model’s individual fairness. Zero-Knowledge Proofs will then be leveraged to verify the group and individual fairness of the model.

**References**

1. Karkkainen, K. and J. Joo. *Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation*. in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021.

2. Buolamwini, J. and T. Gebru. *Gender shades: Intersectional accuracy disparities in commercial gender classification*. in *Conference on fairness, accountability and transparency*. 2018. PMLR.

3. Leslie, D., *Understanding bias in facial recognition technologies.* arXiv preprint arXiv:2010.07023, 2020.

4. Jadhav, A. *How are biometric systems evaluated?* 2024; Available from: <https://www.biometricupdate.com/202405/how-are-biometric-systems-evaluated>.

5. Howard, J.J., et al. *Evaluating proposed fairness models for face recognition algorithms*. in *International Conference on Pattern Recognition*. 2022. Springer.

6. Valdivia, A., J.C. Serrajòrdia, and A. Swianiewicz, *There is an elephant in the room: Towards a critique on the use of fairness in biometrics.* AI and Ethics, 2023. **3**(4): p. 1407-1422.

7. Barclay, I., et al., *Certifying provenance of scientific datasets with self-sovereign identity and verifiable credentials.* arXiv preprint arXiv:2004.02796, 2020.

8. Gong, S., X. Liu, and A.K. Jain. *Mitigating face recognition bias via group adaptive classifier*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

9. Wang, Z., et al. *Towards fairness in visual recognition: Effective strategies for bias mitigation*. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

10. Dehdashtian, S., et al., *Fairness and Bias Mitigation in Computer Vision: A Survey.* arXiv preprint arXiv:2408.02464, 2024.

11. Yadav, C., et al., *FairProof: Confidential and Certifiable Fairness for Neural Networks.* arXiv preprint arXiv:2402.12572, 2024.

12. Fiege, U., A. Fiat, and A. Shamir. *Zero knowledge proofs of identity*. in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. 1987.

13. Yan, T. and C. Zhang. *Active fairness auditing*. in *International Conference on Machine Learning*. 2022. PMLR.

14. Casper, S., et al. *Black-box access is insufficient for rigorous ai audits*. in *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. 2024.

15. Drozdowski, P., et al., *Demographic bias in biometrics: A survey on an emerging challenge.* IEEE Transactions on Technology and Society, 2020. **1**(2): p. 89-103.

16. Yucer, S., et al., *Racial bias within face recognition: A survey.* ACM Computing Surveys, 2024. **57**(4): p. 1-39.

17. Bae, G., et al. *Digiface-1m: 1 million digital face images for face recognition*. in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023.

18. Chandaliya, P.K., et al., *Towards Inclusive Face Recognition Through Synthetic Ethnicity Alteration.* arXiv preprint arXiv:2405.01273, 2024.

19. Jordan, M., J. Lewis, and A.G. Dimakis, *Provable certificates for adversarial examples: Fitting a ball in the union of polytopes.* Advances in neural information processing systems, 2019. **32**.

1. <https://docs.openg2p.org/utilities-and-tools/4sure-verifier> [↑](#footnote-ref-1)
2. <https://docs.mosip.io/1.1.5/biometrics/biometric-sdk> [↑](#footnote-ref-2)
3. According to footnote1, under the considerations for Face match SDK, the OpenG2P documentation mentions the utilisation of a TensorFlow Lite model and states: “The tflite model requires creation and training by the integrating party, demanding specific technical expertise.” [↑](#footnote-ref-3)
4. <https://docs.mosip.io/1.1.5/apis/biometric-sdk-api-specification>. Under the Introduction heading the documentation states: “Mosip as a platform does not have any inbuilt capabilities to handle biometrics. It relies on external components and subsystems to perform all activities pertaining to biometrics. As a platform it defines formats, standards and interfaces for these external components and subsystems.” [↑](#footnote-ref-4)
5. <https://docs.openg2p.org/utilities-and-tools/4sure-verifier> [↑](#footnote-ref-5)
6. <https://docs.mosip.io/1.1.5/biometrics/biometric-sdk> [↑](#footnote-ref-6)
7. <https://docs.mosip.io/1.2.0/modules/id-repository> [↑](#footnote-ref-7)
8. <https://docs.mosip.io/compliance-tool-kit> [↑](#footnote-ref-8)
9. <https://docs.mosip.io/1.2.0/biometrics/abis-api> [↑](#footnote-ref-9)
10. <https://docs.mosip.io/1.2.0/biometrics/abis-api#identify> [↑](#footnote-ref-10)
11. <https://global.itu.edu.tr/docs/librariesprovider33/default-document-library/%C4%B1cao-standarts.pdf?sfvrsn=c9eea239_0> [↑](#footnote-ref-11)
12. <https://skintone.google/get-started> [↑](#footnote-ref-12)
13. <https://huggingface.co/pravsels/synthpar2> [↑](#footnote-ref-13)
14. <https://skintone.google/mste-dataset> [↑](#footnote-ref-14)
15. <https://github.com/serengil/deepface> [↑](#footnote-ref-15)
16. <https://github.com/infinite-pursuits/FairProof> [↑](#footnote-ref-16)