Fair Universal Representations using Adversarial Models

Peter Kairouz [1] **Jiachun Liao** [2] **Chong Huang** [2] **Monica Welfert** [2] **Lalitha Sankar** [2]

Abstract

We present a data-driven framework for learning *fair universal representations* (FUR) that guarantee statistical fairness for any learning task that may not be known *a priori*. Our framework leverages recent advances in adversarial learning to allow a data holder to learn representations in which a set of sensitive attributes are decoupled from the rest of the dataset. We formulate this as a constrained minimax game between an encoder and an adversary where the constraint ensures a measure of usefulness (utility) of the representation. For appropriately chosen adversarial loss functions, our framework precisely clarifies the optimal adversarial strategy against strong information-theoretic adversaries; it also achieves the fairness measure of demographic parity for the resulting constrained representations. We highlight our results for the UCI Adult and UTKFace datasets.

1. Introduction

The use of data-driven machine learning (ML) has recently seen unprecedented success in a variety of automated decision-making systems including facial recognition, natural language processing, mortgage lending, and even parole prediction. The success of these approaches hinges on the availability of large datasets that often include sensitive personal information. It has been shown that models learned from such datasets can inherit societal bias and discrimination patterns (Ladd, 1998; Pedreshi et al., 2008) and learn sensitive features even when they are not explicitly used during training (Song & Shmatikov, 2019). Concerns about the fairness, bias, and privacy of learning algorithms have led to a growing body of research focused on both defining meaningful fairness measures and designing algorithms with such guarantees.

Three distinct approaches have been considered to assure fair ML: in-processing, pre-processing, and post-processing. In-processing approaches are common in the supervised setting where the learning objective is known (e.g., (Dwork et al., 2012; Zhang et al., 2018)); the resulting trained model guarantees fairness for the specific objective. Pre-processing generally produces fair representations tuned for a chosen learning objective (Madras et al., 2018; Edwards & Storkey, 2016; Calmon et al., 2017) while post-processing provides fairness by properly altering decision outputs (Hardt et al., 2016; Hajian et al., 2015; Wei et al., 2020).

Censoring has emerged as a compelling pre-processing approach wherein protected features (e.g., race, gender, and their correlates) are actively decorrelated from the rest of the data to explicitly limit their effect on decisions. Censoring is inspired by information-theoretic privacy methods to limit leakage of sensitive features (Hamm, 2017; Huang et al., 2017; Bertran et al., 2019; Song & Shmatikov, 2019) and can be achieved using generative adversarial networks (GANs) (Goodfellow et al., 2014). Censoring for fairness has largely focused on learning fair predictors (Edwards & Storkey, 2016; Madras et al., 2018; Zhang et al., 2018).

Our Contributions: Taking a preprocessing approach, the main contribution of this work is to use censoring to generate fair representations (FRs) that are *universal*. These are representations from which the sensitive features have been actively decoupled and can be universally used for a variety of *a priori* unknown learning tasks. We formally define demographic parity (DemP) for representations and show that our fair universal representation (FUR) framework assures DemP group fairness for all downstream predictions.

There has been recent work on using adversarial methods to generate *transferable* fair representations (Madras et al., 2018); our approach, while similar in philosophy, goes a step further by enforcing a *hard* distortion constraint that allows better control of the learned representations, and therefore, better downstream utility guarantees. Algorithmically, we showcase how Lagrange penalty methods (Lillo et al., 1993) can help enforce the hard constraint in a GAN-setting[1].

Our most important contribution is in illustrating the utility

[1]Google Research [2]Arizona State University. Correspondence to: Monica Welfert <mwelfert@asu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1] Recently TensorFlow updated its package to allow enforcing hard constraints (Google) using a similar approach.

of FURs for two publicly available datasets, namely, the UCI Adult (Kohavi, 1996) and the UTKFace (Zhang et al., 2017). Our visual results demonstrate our success in creating high quality representations that increasingly erase the sensitive attributes with decreasing fidelity requirements. In contrast to state-of-the-art (Madras et al., 2018; Edwards & Storkey, 2016; Zhang et al., 2018), our framework is the first to include non-binary sensitive attributes, multiple downstream tasks, as well as hard distortion constraints. Our results show that one can still learn high accuracy DemP (and even equal opportunity) fair classifiers from DemP FURs. Finally, our results straddle a wide range of values for DemP, thereby including perfect fairness, in contrast to the above works.

We set up the problem and review fairness measures in Section 2. In Section 3, we formalize our framework, define censored and fair representations, and highlight the theoretical guarantees of this approach. We showcase the performance of the FUR framework on the UCI Adult and UTKFace datasets in Section 4. Proofs, algorithm, and deep learning architectures for the datasets are in the Appendix.

2. Preliminaries

Consider a dataset $\mathcal{D}$ with n entries where each entry is a random tuple $(S, X, Y) \in \mathcal{S} \times \mathcal{X} \times \mathcal{Y}$ where S, X, and Y are sensitive, non-sensitive, and target (non-sensitive) features, respectively, and $\hat{Y} \in \mathcal{Y}$ is a predictor of Y. Note that S and Y can be a collection of features or labels (e.g., S can be gender, race, sexual orientation, or a combination of these, while Y could be age, facial expression, etc.); we use the term variable to denote both single and multiple features/labels. Instances of X, S, and Y are denoted by x, s and y, respectively. The entries (X, S, Y) of $\mathcal{D}$ are independent and identically distributed (i.i.d.) according to $P(X, S, Y)$. We emphasize that Y represents a set of downstream ML tasks and is not used to create FRs.

Algorithmic fairness measures try to guarantee that, for a specific target Y, the prediction of a ML model is accurate with respect to (*w.r.t.*) Y but unbiased *w.r.t.* the sensitive S. While more than two dozen measures for fairness have been proposed, two oft-used fairness measures are DemP and equalized odds (EO) (and variants thereof). DemP ensures complete independence between the prediction of the target $\hat{Y}$ and the sensitive S; this notion of fairness favors utility the least, especially when Y and S are correlated (Hardt et al., 2016). EO enforces this independence conditioned on Y, thereby ensuring equal rates for true and false positives (when Y is binary) for all demographics. We now define DemP and EO formally (for binary S and Y as originally introduced). These can be generalized to the non-binary setting, and we do so later for FRs.

Definition 2.1 ((Hardt et al., 2016)). A predictor $f(S, X) = \hat{Y}$ satisfies

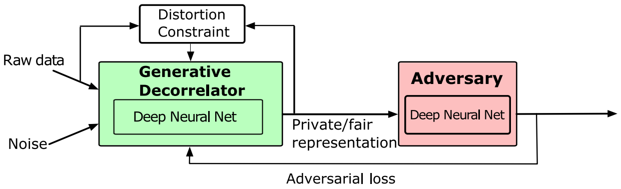


Figure 1. Generative adversarial model for censoring/fairness.

- demographic parity (DemP) w.r.t. S, if $\hat{Y} \perp S$, i.e.,

$$Pr(\hat{Y}=1|S=1)=Pr(\hat{Y}=1|S=0) \quad (1)$$

- equalized odds (EO) *w.r.t.* (S, Y), if $\hat{Y} \perp S|Y$, for $y \in \{0, 1\}$:

$$Pr(\hat{Y}=1|S=1, Y=y)=Pr(\hat{Y}=1|S=0, Y=y). \quad (2)$$

In the following section, we present our FUR framework.

3. FURs via Generative Adversarial Models

Formally, the FUR model consists of an encoder and an adversary, as shown in Figure 1. The goal of the encoder $g : \mathcal{X} \times \mathcal{S} \to \mathcal{X}_r$ is to actively eliminate the dependence between S and X while that of the adversary $h : \mathcal{X}_r \to \mathcal{S}$ is to infer S. In general, $g(X, S)$ is a randomized mapping that outputs a representation $X_r = g(X, S)$. Note that S may not always be available to the curator; however, it will always affect the design of g via the adversarial training process. For brevity, we henceforth write $g(\cdot)$ to include both possibilities (just X or (X, S) as inputs). On the other hand, the role of the adversary is captured via $h(X_r)$, which is the adversarial decision rule in inferring the sensitive variable S as $\hat{S} = h(X_r = g(\cdot))$ from the representation $g(\cdot)$. In general, the hypothesis h can be a *hard decision rule* under which $h(g(\cdot))$ is a direct estimate of S or a *soft decision rule* under which $h(g(\cdot)) = P_h(\cdot|g(\cdot))$ is a distribution over $\mathcal{S}$.

To quantify the adversary's performance, we use a loss function $\ell(h(g(X = x, S = s)), S = s)$ defined for every pair (x, s). Thus, the adversary's expected loss *w.r.t.* X and S is $L(h, g) \triangleq \mathbb{E}[\ell(h(g(\cdot)), S)]$, where the expectation is taken over $P(X, S)$ and the randomness in g and h. To ensure utility, we introduce a constraint on the fidelity of X_r via a distortion function $d(x_r, x)$, which measures the goodness of $X_r = x_r$ *w.r.t.* $X = x$. We ensure statistical utility by constraining the average distortion $\mathbb{E}[d(g(\cdot), X)]$, where the expectation is taken over $P(X, S)$ and the randomness in g.

3.1. FUR: Framework and Theoretical Results

Generating an FR X_r requires learning an encoder g that guarantees both censoring (i.e., it is difficult for the adversary to learn S from X_r) and utility (g guarantees bounded distortion of X). For a fixed g, the adversary learns a (potentially randomized) function h that minimizes its expected loss in inferring S, or equivalently maximizes the negative expected loss. This leads to a constrained minimax game between the encoder and the adversary given by

$$\min_{g(\cdot)} \max_{h(\cdot)} -\mathbb{E}[\ell(h(g(\cdot)), S)], \text{ s.t. } \mathbb{E}[d(g(\cdot), X)] \leq D, \quad (3)$$

where $D \geq 0$ determines the distortion constraint on X_r. The optimization in (3) highlights that the input to g depends on whether the curator has access to both (X, S) or just X. Having access to both (X, S) in general will yield a better decorrelator (e.g., see Section 4.1 for the UCI dataset). Finally, without the constraint in (3), the optimal $X_r = g(\cdot) \perp S$. One can approximate this in practice via arbitrarily large distortions as we show in Theorem 3.4; as a setup to these results, we first define censoring and fairness for representations. Our censoring definition clarifies the representation that best limits an adversary from inferring S. We then define DemP for FRs; we combine the two definitions to show how and when adversarial learning can help ensure demographic parity.

Definition 3.1 (Censored Representations)**.** A representation X_r of X is censored *w.r.t.* the sensitive features S against a learning adversary $h(\cdot)$, whose performance is evaluated via a loss function $\ell(h(X_r), S)$, if for an optimal adversarial strategy $h^* = \text{argmin}_h \mathbb{E}[\ell(h(X_r), S)]$,

$$\mathbb{E}[\ell(h^*(g(\cdot)), S)] \leq \mathbb{E}[\ell(h^*(X_r), S)], \quad (4)$$

where $g(\cdot)$ is any (randomized) function of X (or (X, S)) and the expectation is over h, g, X, and S.

The above definition suggests that the best censored representation X_r is the least informative about S to an adversary whose inferential action is captured by a loss function $\ell(\cdot, \cdot)$, i.e., the average loss is the worst for X_r than for any other arbitrary function $g(\cdot)$. While the comparison in (4) is *w.r.t.* the best $h^*(X_r)$ for X_r, choosing the optimal $h(\cdot)$ for any $g(\cdot)$ will only serve as a lower bound to the left side of (4).

We now define DemP for representations; we then prove that a DemP FR X_r guarantees that any downstream algorithm using X_r satisfies DemP *w.r.t.* S.

Definition 3.2 (Demographically Fair Representations)**.** For $(X, S) \in \mathcal{X} \times \mathcal{S}$, a representation $X_r = g(X, S) \in \mathcal{X}_r$ satisfies demographic parity *w.r.t.* S if for any $x_r \in \mathcal{X}_r$ and $s, s' \in \mathcal{S}$

$$Pr(X_r = x_r | S = s) = Pr(X_r = x_r | S = s') \quad (5)$$

where $g : \mathcal{X} \times \mathcal{S} \to \mathcal{X}_r$ is a randomized function.

Theorem 3.3 (Fair Learning via Fair Representation)**.** *If $X_r = g(X, S)$ satisfies DemP w.r.t. S, then any algorithm $f : \mathcal{X}_r \to \mathcal{Y}$ satisfies DemP w.r.t. S.*

Proof of Theorem 3.3 (see Appendix A) follows from a direct application of the data-processing inequality for mutual information since $(X, S) - X_r - Y$ form a Markov chain.

One simple approach to obtain a fair/censored representation X_r is by choosing $X_r = N$ where $N \perp (X, S)$. However, such an X_r has no utility. Since X_r has to ensure utility, there is a tradeoff between guaranteeing fairness/censoring and achieving a desired level of utility as formalized below.

Theorem 3.4. *For sufficiently large distortion bound D, (3) yields a universal representation X_r censored w.r.t. S.*

The proof follows by observing that for sufficiently large D, X_r can be arbitrarily noisy, reducing (3) to an unconstrained optimization. For this X_r with $h^* = \text{argmin}_h \mathbb{E}[\ell(h(X_r), S)]$,

$$\mathbb{E}[\ell(h^*(X_r), S)] = -\min_{g(\cdot)} \max_{h(\cdot)} -\mathbb{E}[\ell(h(g(\cdot)), S)] \quad (6)$$

$$\geq \mathbb{E}[\ell(h^*(g(\cdot)), S)], \quad (7)$$

thus satisfying Definition 3.1.

A predominant approach in the literature in the context of fair representations is to explicitly include the intended classification/prediction task, i.e., design representations that guarantee DemP for the specific task (Madras et al., 2018; Edwards & Storkey, 2016; Zhang et al., 2018). The FUR formulation in (3) can be extended to include this by adding an additional term to the objective that ensures accuracy in learning Y. The resulting minimax game is

$$\min_{\tilde{g}(\cdot), f(\cdot)} \max_{h(\cdot)} -\mathbb{E}[\ell(h(\tilde{g}(\cdot)), S)] + \lambda \mathbb{E}[\ell'(f(\tilde{g}(\cdot)), Y)],$$
$$\text{s.t.} \quad \mathbb{E}[d(\tilde{g}(\cdot), X)] \leq D, \quad (8)$$

where $f(\cdot)$ is a classifier for a target Y, $\lambda > 0$, and $\tilde{g}(\cdot)$[2] and $h(\cdot)$ are the encoder and the adversarial classifier, respectively, as in (3). Note that the loss functions $\ell(\cdot)$ and $\ell'(\cdot)$ can be different. The setup in (8) involves an additional term ensuring fair classification and is, thus, a more constrained optimization than the FUR framework; in fact, we recover the FUR setup with $\lambda = 0$. However, even while generating intermediate representations $g(\cdot)$, (8) is primarily intended to design *fair classifiers*, and therefore, requires knowing the intended tasks on Y. In contrast, our FUR framework allows generating DemP FRs X_r that in turn guarantee DemP fairness to all downstream tasks on any subset of Y.

3.2. Data-driven FUR

We propose a data-driven version of the FUR framework that learns a generative decorrelator $g(X; \theta_p)$, parameterized by θ_p, from an n-sample dataset $\mathcal{D} = \{(x_{(i)}, s_{(i)})\}_{i=1}^n$. This model takes X (or (X, S)) as input and outputs X_r. In the training phase, the data holder learns the optimal parameters θ_p by competing against a *computational adversary*: a classifier modeled by a neural network $h(g(X; \theta_p); \theta_a)$ that is parameterized by θ_a. In the evaluation phase, we use the

[2] In general, $\tilde{g}(\cdot)$ can be a function of both X and S; the dependence on S is implicit when S is not directly available.

accuracy of a classifier learned using X_r as a measure of goodness of the representations and compute the empirical DemP and EO to evaluate the fairness guarantees. For a fixed h and g, binary S and ℓ as log-loss, the adversary's *empirical loss* using cross entropy is given by

$$L_n(\theta_p, \theta_a) = -\frac{1}{n}\sum_{i=1}^{n} s_{(i)} \log h(g(x_{(i)};\theta_p);\theta_a) + (1-s_{(i)})\log(1-h(g(x_{(i)};\theta_p);\theta_a)). \tag{9}$$

The optimal model parameters (θ_p, θ_a) are then solutions of

$$\min_{\theta_p}\max_{\theta_a} -L_n(\theta_p,\theta_a), \text{ s.t. } \frac{1}{n}\sum_{i=1}^{n} d(g(x_{(i)};\theta_p), x_{(i)}) \leq D. \tag{10}$$

It is crucial to note that the *hard* distortion constraint in (10) makes it different from what has been extensively studied in the literature. To incorporate the distortion constraint, we use the *penalty method* (Lillo et al., 1993) that replaces (10) by a series of unconstrained optimization problems by adding a penalty to the objective as detailed in Appendix B.

4. Illustration of Results

We apply our FUR framework to two real-world datasets, namely, UCI Adult (Kohavi, 1996) and UTKFace (Zhang et al., 2017), briefly described below. For both datasets, we restrict the architecture of h, g, and the downstream predictive models to neural networks.

(i) The UCI Adult dataset (Kohavi, 1996) consists of 10 categorical and 4 continuous features and is used to predict a binary salary label (1: salary $>$ 50k or 0: salary $\leq$ 50k). We choose gender or the tuple (gender, relationship) as the sensitive S, the remaining features except salary as non-sensitive X (Table 1 in the Appendix lists all features), and salary as the target Y.

(ii) The UTKFace dataset (Zhang et al., 2017) consists of more than 20k 200×200 color images of faces labeled by age, ethnicity, and gender. Individuals in the dataset have ages from 0 to 116 years and belong to 5 ethnicities: White, Black, Asian, S. Asian (Indian), and others (includes Hispanic, Latino and Middle Eastern). We set gender as S, image as X, and age and ethnicity as two target labels Y, and restrict the data to images for ages between 10 and 65.

We use the accuracy of predicting S as the measure of censoring. We evaluate the fairness guarantees of X_r by computing the DemP obtained on tasks using Y. To this end, we compute the following maximal difference as a proxy for DemP in Definition 2.1 (includes non-binary Y and S):

$$\Delta_{\text{DemP}}(y) = \max_{s,s'\in\mathcal{S}} |P(\hat{Y}=y|S=s) - P(\hat{Y}=y|S=s')| \tag{11}$$

with smaller values of $\Delta_{\text{DemP}}(y)$ suggesting better DemP fairness guarantees. For binary Y, $\Delta_{\text{DemP}}(y)$ in (11) simplifies to a single value that we denote as Δ_{DemP}. In our experiments, we use the empirical frequencies to estimate $P(\hat{Y}=y|S=s)$ for a chosen (y,s). We illustrate both censoring and fairness results for the abovementioned datasets in the following subsections. Experimental and model details are in Appendix C.

4.1. Illustration of Results for UCI Adult Dataset

For the UCI Adult dataset with both categorical and continuous features as shown in Table 1 in the Appendix, we consider two cases:
(i) Case I: binary S by choosing 'gender' as sensitive feature
(ii) Case II: non-binary S by considering both 'gender' and 'relationship' as sensitive.
For both cases, 'salary' is the binary target $Y\in\{0,1\}$, with $Y=1$ denoting salary $>50K$. Since the two values for $\Delta_{\text{DemP}}(y)$ in (11) are the same for binary Y, we write Δ_{DemP} when illustrating results. We illustrate Case I below; Case II is relegated to Appendix C.1.1 for reasons of space.

Case I: Binary Sensitive Feature.

Figure 2 illustrates the censoring and fairness performance of the FR X_r for the UCI dataset. For censoring, the performance is evaluated via the tradeoff between the classification accuracies of salary (utility of X_r) and gender (censoring of S). Note that salary accuracy is evaluated as a downstream task via a separately learned classifier that uses X_r while gender accuracy is a measure of performance of the neural network adversary h in the FUR model. We evaluate fairness via the tradeoff between salary accuracy and Δ_{DemP}. We consider two possible inputs to the encoder $g(\cdot)$ in (3), i.e., only X or both (X,S).

From Figure 2a, the baseline[3] salary and gender accuracies for the UCI dataset are about 84.5% and 85%, respectively. Further, for the FUR X_r and downstream $\hat{Y}$:
(i) the smallest gender accuracy achievable is about 66%, 20% below its baseline, while the lowest salary accuracy is about 82%, 2.5% below its baseline. Since the likelihood of a male in the original test data is 66%, with increasing distortion, the FUR gender accuracy is as good as a random guess, i.e., the generated X_r hides gender effectively while maintaining high salary accuracy.
(ii) For the same gender accuracy, using both (X,S) appears useful only for high utility regime (salary accuracy $\geq 83\%$).

From Figure 2b, we observe the following:
(i) salary classification accuracy and Δ_{DemP} have an approximately affine relationship, and when $\Delta_{\text{DemP}}\approx 0$, the salary accuracy is $\geq 79\%$, i.e., the FUR framework is effective in

[3] The baseline performances are the salary and gender accuracies as well as Δ_{DemP} obtained from the original uncensored test dataset.

approaching perfect DemP with a small reduction in utility; (ii) the FURs X_r generated from either X or (S, X) lead to similar fairness guarantees. For $\Delta_{\text{DemP}} = 0.06$, state-of-the-art approaches in (Edwards & Storkey, 2016) and (Madras et al., 2018) achieve 2% and 2.5% higher salary accuracy than ours, respectively; however, our approach is distinct in achieving $\Delta_{\text{DemP}} \approx 0$ with salary accuracy $\geq 79\%$.

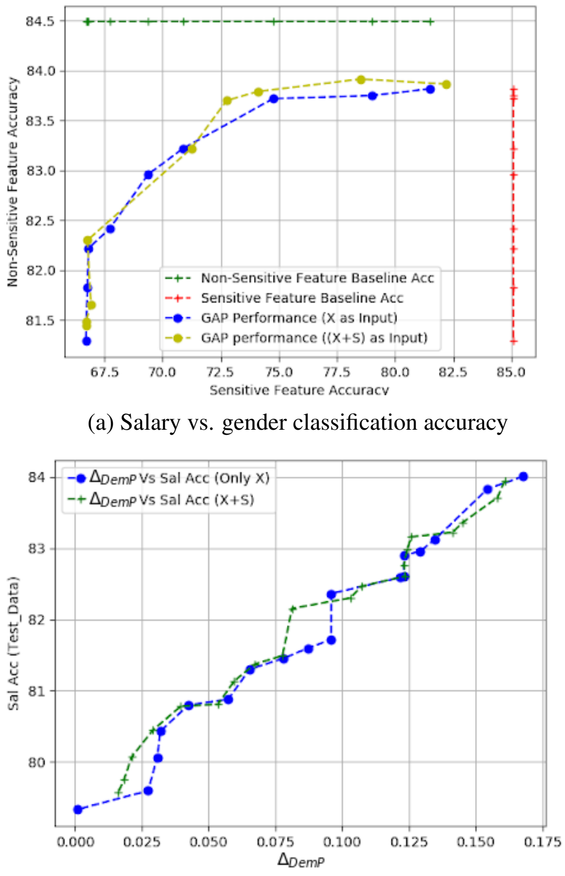
From Figure 2a, we see that gender accuracy saturates at 67% while achieving a salary accuracy of at least 81% for a specific value of distortion bound D, and therefore, test distortion; in turn, this choice of D corresponds in Figure 2b to $\Delta_{\text{DemP}} \approx 0.06$. Further reducing Δ_{DemP} requires further increasing D, thus lowering the salary accuracy to 79% for $\Delta_{\text{DemP}} \approx 0$. This is because classification accuracy captures an average measure of correctness and is dominated by the performance over the majority class. On the other hand, Δ_{DemP} captures the difference in performance of the intended classifier on each of the two classes. Thus, enforcing fairness via Δ_{DemP} reduces salary accuracy thereby highlighting the tradeoff between guaranteeing fairness and utility.

We can also evaluate the fairness performance of the generated X_r by using the EO measure in Definition 2.1. Thus, for $Y \in \{0, 1\}$ where $Y = 1$ when salary $> 50K$, $S \in \{0, 1\}$ (female:1 and male:0), and $\hat{Y} \in \{0, 1\}$, we write $\Delta_{\text{EO}}(y), \forall y \in \{0, 1\}$ as:

$$\Delta_{\text{EO}}(y) \triangleq \left| P(\hat{Y}=y|S=0, Y=y) - P(\hat{Y}=y|S=1, Y=y) \right| . \tag{12}$$

Note that for binary Y, as is the case here, (12) is the same as the definition of EO in (2). From Figure 3, which plots salary accuracy vs. DemP or EO measures of fairness, we observe that while the salary accuracy is above 82.4%, the values of $\Delta_{\text{EO}}(1)$ and $\Delta_{\text{EO}}(0)$ decrease to 0.0007 and 0.0254, respectively. To understand the significance of these results, we compare against the state-of-the-art in (Madras et al., 2018), wherein fair salary classifiers for both DemP and EO measures, referred to as LAFTR-DP[4] and LAFTR-EO, respectively, are learned for the UCI dataset. For the LAFTR-DP, the authors also compute the resulting EO of the DemP classifier. As a preamble to the following comparisons, we note that fair predictors, trained on specific tasks, will do at least as well as the same predictors learned on FRs.

We make the following observations: (i) when $\Delta_{\text{EO}}(1) + \Delta_{\text{EO}}(0) = 0.04$[5], our salary accuracy is 1.3% smaller than that achieved by LAFTR-DP (cf. Figure 2b in (Madras et al., 2018)), but our minimal achieved value of $\Delta_{\text{EO}}(1) + \Delta_{\text{EO}}(0)$ is only 72% of that achieved by LAFTR-DP and is the same as the value achieved by LAFTR-EO, which uses EO as the fairness metric to train a salary classifier; (ii) the decrease of $\Delta_{\text{EO}}(1) + \Delta_{\text{EO}}(0)$ is even larger than Δ_{DemP}. That is, even though the representation is generated to satisfy DemP, it can also provide competitive downstream EO fairness guarantees. This, in turn, justifies the rationality of generating fair representations under DemP.



(a) Salary vs. gender classification accuracy

(b) Salary classification accuracy vs. Δ_{DemP}

Figure 2. Results for UCI Adult: Case I. In Figure 2a, the green and red lines denote the baseline performances for the target Y (salary) and sensitive S (gender), respectively; in Figure 2b, the value of Δ_{DemP} for the original test data is 0.2. In both plots, each point corresponds to a specific value of achieved test distortion; for Figures 2a and 2b, the achieved test distortion for the blue points ranges over $(0.69, 4.1)$ and $(0.69, 4.4)$, respectively, with decreasing distortion from left to right for each plot. The achieved test distortion for the yellow-green points ranges over $(0.87, 4.2)$ and $(0.87, 4.9)$, respectively.

4.2. Illustration of Results for UTKFace Dataset

In the UTKFace dataset, the face images are the non-sensitive X. We choose 'gender' as the sensitive S; focusing on multiple downstream tasks, we consider both ethnicity classification and age regression, for which we choose 'ethnicity' or 'age' as the target variable Y, respec-

[4] Learned Adversarially Fair and Transferable Representations (LAFTR)

[5] (Madras et al., 2018) introduced an EO measure as $\Delta_{\text{EO}} \triangleq \Delta_{\text{EO}}(1) + \Delta_{\text{EO}}(0)$.

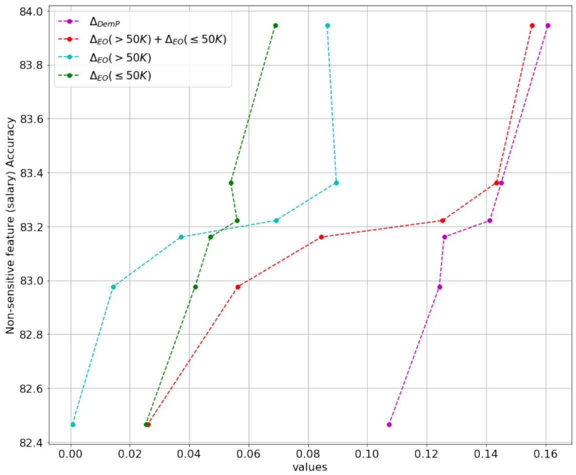
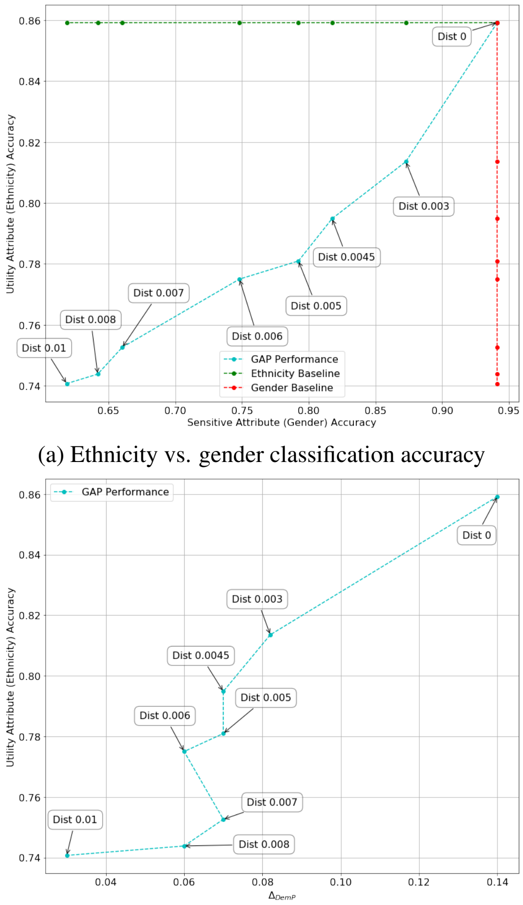


Figure 3. Evaluation of equalized odds fairness metric under Case I for the UCI Adult dataset. The EO measures $\Delta_{\mathrm{EO}}(1)$ and $\Delta_{\mathrm{EO}}(0)$ are defined in (12). The red curve plotting $\Delta_{\mathrm{EO}} = \Delta_{\mathrm{EO}}(1) + \Delta_{\mathrm{EO}}(0)$ matches Δ_{EO} in Figure 2(b) of (Madras et al., 2018). Each point corresponds to a specific value of achieved test distortion ranging over $(0.59, 2.01)$, with distortion decreasing from left to right for each plot.

tively. We detail the results for ethnicity classification in this section, while those for age regression can be found in Appendix C.2.1. The support of Y for ethnicity classification is $\mathcal{Y} = \{\text{White, Black, Asian, Indian}\}$. We use the maximum of the DemP measure, defined in (11), over the support Y, i.e., $\Delta_{\mathrm{DemP}} = \max_{y \in \mathcal{Y}} \Delta_{\mathrm{DemP}}(y)$, as the achieved fairness level.

Figure 4a shows the tradeoffs between gender and ethnicity classification accuracies. While gender accuracy is about 62% and decreases about 35% from the baseline, the ethnicity classification accuracy is above 74% and only decreases 14% from its baseline performance. Note that in the original test data, the highest marginal probabilities for gender and ethnicity are 54.6% (likelihood of male) and 43.2% (likelihood of White), respectively. Therefore, gender accuracy is better than a random guess by only 7.4% while ethnicity accuracy is better than a random guess by 30.8%, i.e., the generated X_r hides gender information well while maintaining ethnicity, illustrating that distortion constrained FRs X_r can guarantee utility for this task.

Figure 4b illustrates the tradeoff between the utility measure and Δ_{DemP} of the generated X_r in ethnicity classification. We observe that while achieving about 86% of the baseline classification accuracy, the Δ_{DemP} is reduced to 0.03, which is 20% of the $\Delta_{\mathrm{DemP}} = 0.14$ in the original test data. Table 2 in the Appendix shows the decrease of Δ_{DemP} for every ethnicity as the distortion increases. Finally, we visually illustrate the FR images in Figure 9 in the Appendix and discuss the effect of distortion on learning FRs.



(a) Ethnicity vs. gender classification accuracy

(b) Ethnicity classification accuracy vs. Δ_{DemP}

Figure 4. Ethnicity classification accuracy vs. gender classification and Δ_{DemP} for the UTKFace dataset. In Fig. 4b, the x-axis is the maximal value of DemP in (11) over the four ethnicities and 'dist' indicates the per pixel distortion.

5. Conclusion

We have introduced an adversarial learning framework with verifiable guarantees for learning generative models that can create censored and fair universal representations for datasets with known sensitive features. The novelty of our approach is in producing representations that are fair with respect to the sensitive features for any *a priori* unknown downstream learning task. We have shown that our FUR framework allows the data holder to learn the fair encoding scheme (a randomized mapping that decorrelates the sensitive and non-sensitive features) directly from the dataset without requiring access to dataset statistics. A promising area to expand and explore this framework is for healthcare data; the challenge here is in learning FRs when sensitive features such as race may need to be both censored and used appropriately in predictive tasks.

References

Abadi, M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint:1603.04467*, 2016.

Bertran, M., Martinez, N., Papadaki, A., Qiu, Q., Rodrigues, M., Reeves, G., and Sapiro, G. Adversarially learned representations for information obfuscation and inference. In *Proc. ICML*, pp. 614–623, 2019.

Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., and Varshney, K. R. Optimized pre-processing for discrimination prevention. In *NeurIPS*, pp. 3992–4001, 2017.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proc. Innovations in Th. CS*, pp. 214–226, 2012.

Eckstein, J. and Yao, W. Augmented lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Res. Rep.*, 32, 2012.

Edwards, H. and Storkey, A. J. Censoring representations with an adversary. In *ICLR*, pp. 1–14, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proc. NeurIPS*, pp. 2672–2680, 2014.

Google. https://github.com/google-research/tensorflow_constrained_optimization/blob/master/README.md.

Hajian, S., Domingo-Ferrer, J., Monreale, A., Pedreschi, D., and Giannotti, F. Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery*, 29(6): 1733–1782, 2015.

Hamm, J. Minimax filter: learning to preserve privacy from inference attacks. *J. Mach. Learn. Research*, 18(1): 4704–4734, 2017.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *Proc. NeurIPS*, pp. 3315–3323, 2016.

Huang, C., Kairouz, P., Chen, X., Sankar, L., and Rajagopal, R. Context-aware generative adversarial privacy. *Entropy*, 19(12), 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint:1412.6980*, 2017.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, volume 96, pp. 202–207, 1996.

Ladd, H. F. Evidence on discrimination in mortgage lending. *Journal of Economic Perspectives*, 12(2):41–62, 1998.

Lillo, W. E., Loh, M. H., Hui, S., and Zak, S. H. On solving constrained optimization problems with neural networks: A penalty method approach. *IEEE Trans. Neural Nets.*, 4 (6):931–940, 1993.

Madras, D., Creager, E., Pitassi, T., and Zemel, R. Learning adversarially fair and transferable representations. In *Proc. ICML*, pp. 3384–3393, 2018.

Pedreshi, D., Ruggieri, S., and Turini, F. Discrimination-aware data mining. In *14th ACM SIGKDD*, pp. 560–568, 2008.

Song, C. and Shmatikov, V. Overlearning reveals sensitive attributes. *arXiv:1905.11742*, 2019.

Wei, D., Ramamurthy, K. N., and Calmon, F. Optimized score transformation for fair classification. In *Proc. AISTATS*, pp. 1673–1683, 2020.

Zhang, B. H., Lemoine, B., and Mitchell, M. Mitigating unwanted biases with adversarial learning. In *Proc. IEEE/ACM AIES*, 2018.

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *Proc. CVPR*, 2017.

A. Proof of Theorem 3.3

A demographically fair encoder $g(X,S)$ ensures that $X_r \perp S$, i.e., the mutual information $I(S;X_r)=0$. Further, the downstream learning algorithm acts only on X_r to predict $\hat{Y}$; thus, $(S,X)-X_r-\hat{Y}$ form a Markov chain. From the data processing inequality and non-negativity of mutual information, we have $0 \leq I(S;\hat{Y}) \leq I(S;X_r)=0$, i.e., S is independent of $\hat{Y}$; thus, from Definition 2.1, $\hat{Y}$ satisfies DemP w.r.t. S. Finally, from the chain rule and non-negativity of mutual information, we have $I(X_r;S_t)=0$ for any $S_t \subset S$, i.e., $\hat{Y}$ satisfies DemP w.r.t. any subset of sensitive features S_t.

B. Alternate Minimax Algorithm

Algorithm 1 Alternating minimax FUR algorithm

```
Input: dataset $\mathcal{D}$, distortion parameter $D$, # of decorrelator iterations $T$, # of adversary iterations $J$ for each round of decorrelator update, minibatch size $M$
Output: Optimal generative decorrelator parameter $\theta_p$
function ALTERNATE MINIMAX($\mathcal{D}, D, T, J, M$)
    Initialize decorrelator parameter $\theta_p^1$, adversary parameter $\theta_a^1$, and step size $\eta_1$
    for $t=1,...,T$ do
        Random minibatch of $M$ datapoints $\{x_{(1)},...,x_{(M)}\}$ drawn from full dataset
        Generate $\{\hat{x}_{(1)},...,\hat{x}_{(M)}\}$ via $\hat{x}_{(i)}=g(x_{(i)};\theta_p^t)$
        Apply update rule for step size $\eta_t$
        Set $\omega_a^1$ = $\theta_a^t$
        for $j=1,...,J$ do
            Update the adversary parameter $\theta_a^{t+1}$ by stochastic gradient ascent for epoch $j$
                $$\omega_a^{j+1}=\omega_a^j+\eta_t\nabla_{\omega_a^j}\frac{1}{M}\sum_{i=1}^{M}-\ell(h(\hat{x}_{(i)};\omega_a^j),s_{(i)}), \quad \eta_t>0$$
        end for
        Set $\theta_a^{t+1}$ = $\omega_a^{J+1}$
        Compute the descent direction $\nabla_{\theta_p^t}L_m(\theta_p^t,\theta_a^{t+1})$, where $L_m(\theta_p^t,\theta_a^{t+1})$ is defined in (9) for $n=m$
        Perform line search along $\nabla_{\theta_p^t}L_m(\theta_p^t,\theta_a^{t+1})$ and, for $\ell(\theta_p^t,\theta_a^{t+1})$ set as the objective in (13) for $n=m$, update
                $$\theta_p^{t+1}=\theta_p^t-\eta_t\nabla_{\theta_p^t}\ell(\theta_p^t,\theta_a^{t+1})$$
    end for
    return $\theta_p^{T+1}$
end function
```

Algorithm 1 details the steps used to learn the FUR model in a data-driven manner. To incorporate the distortion constraint, we use the *penalty method* (Lillo et al., 1993) to replace the constrained optimization problem by adding a penalty to the objective function. This is done via a penalty parameter ρ_t that captures a measure of violation of the constraint at the t^{th} iteration. The constrained optimization problem of g is then approximated by a *series of unconstrained optimization problems* with an objective

$$-L_n(\theta_p,\theta_a)+\rho_t(\max\{0,\frac{1}{n}\sum_{i=1}^{n}d(g(x_{(i)};\theta_p),x_{(i)})-D\})^2, \tag{13}$$

where the penalty coefficient ρ_t decreases with the number of iterations t. We start with a large value of ρ_t to enforce distortion from the outset and decrease ρ_t in exponential steps with respect to the number of training epochs. Such a decrease allows enforcing a smaller penalty when the model is closer to convergence. We also vary the learning rate η_t over training epochs as follows: we pick a small value of η_t at the beginning and compare the relative values of the adversarial loss and the average distortion. We adjust the initial η_t so that the adversarial loss and the distortion penalty values are on a similar scale in the first few epochs during training. When the algorithm terminates, we check the average distortion and manually fine tune the initial η_t and the update rule to make sure the distortion is within bounds after termination.We note that both

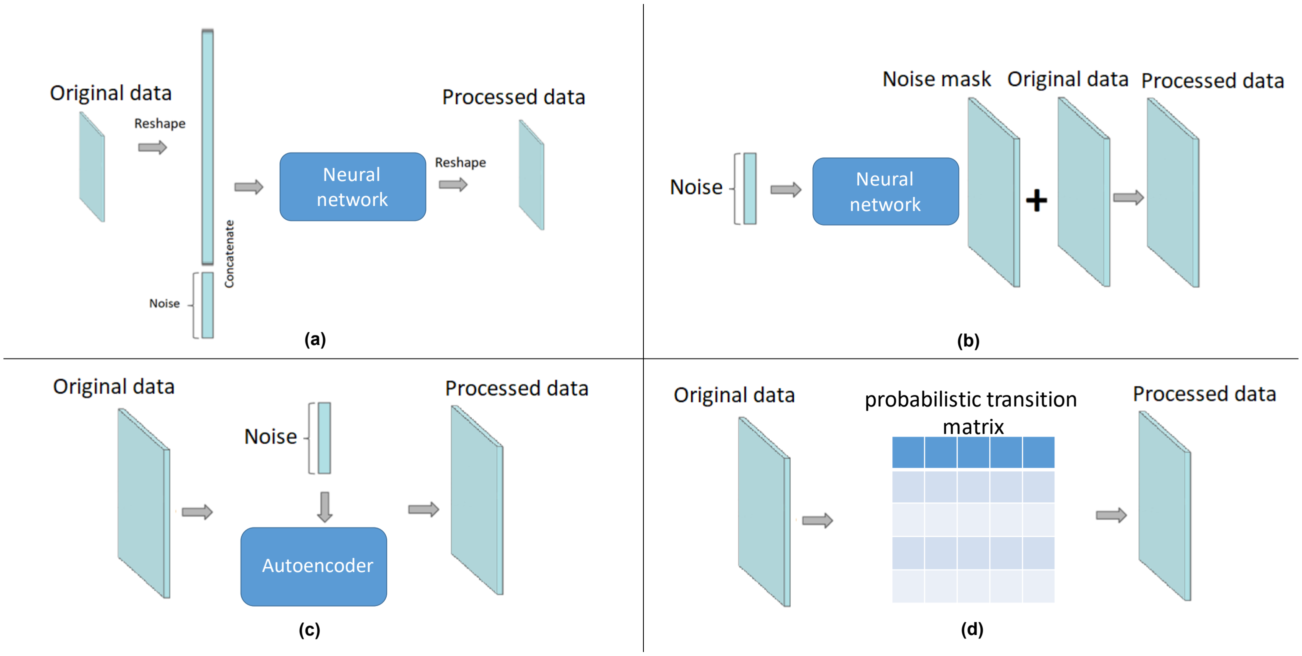


Figure 5. Different architectures of the decorrelator/encoder in the FUR framework.

the augmented Lagrangian and the penalty methods have similar performance in practice; we chose the penalty method but our results can also be obtained with the augmented Lagrangian method (Eckstein & Yao, 2012). Finally, we note that one can easily generalize (9) to the multi-class setting (non-binary S) using the softmax function.

C. Experimental Details and Additional Results

This section contains additional results for and the architectural details of the UCI Adult and UTKFace dataset experiments presented in the main paper. For the two publicly available datasets, the FUR architectures we consider capture two of the four possible approaches to decorrelating the data (X, S). These approaches are illustrated in Figure 5 and include: the feedforward neural network decorrelator (FNND) in Figure 5-(a), the transposed convolution neural network decorrelator (TCNND) in Figure 5-(b), the noisy autoencoder decorrelator (NAED) in Figure 5-(c), and the probability matrix model (PMM) in Figure 5-(d).

The FNND architecture uses a feedforward multi-layer neural network to combine the low-dimensional random noise and the original data (i.e., X or (X, S)) together. The TCNND generates high-dimensional noise from low-dimensional random noise by using a multi-layer transposed convolution neural network, and then, adds it to the original data to produce the representation X_r. The NAED uses the encoder of an autoencoder to generate a lower-dimensional feature vector of the original data and adds independent random noise to each element of the feature vector. The decoder of the autoencoder reconverts the noisy feature vector to generate the processed data X_r. Finally, for purely discrete X and S, X_r can be generated using a probability transition matrix; such a matrix can be learned from the data and is then used to map the entries of the original data to any one of the other entries using the corresponding row of the probability matrix.

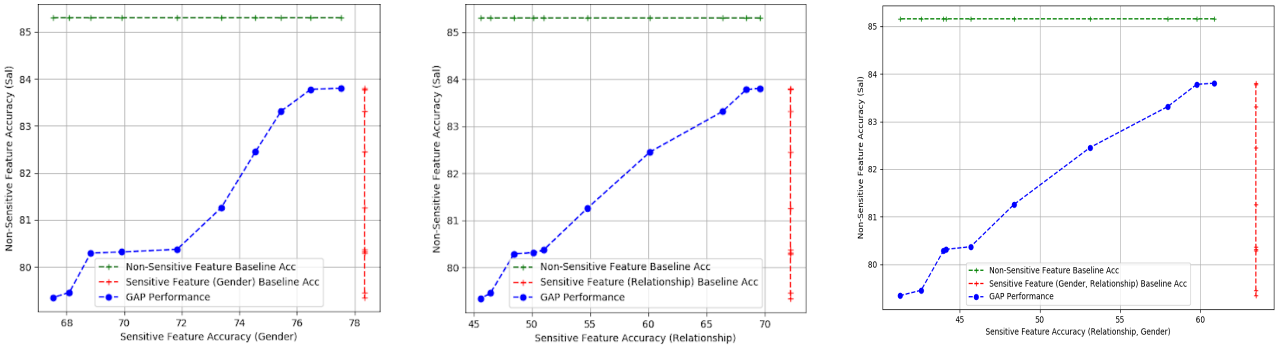
For the UCI Adult dataset, we use the FNND architecture, while for the UTKFace dataset, we consider the NAED architecture. Given that the datasets we consider are either continuous valued or have a mix of continuous and discrete valued features (UCI), we do not use the PMM approach in our experiments. Finally, we train our models based on the data-driven version of the FUR formulation presented in Section 3.2 using TensorFlow (Abadi et al., 2016).

C.1. UCI Adult Dataset Details

Each sample in the UCI Adult dataset has both continuous and categorical features. Table 1 lists all the considered features. We perform a one-hot encoding on each categorical feature in (S, X) and store the mapping function from the one-hot

Table 1. Features of the UCI Adult dataset.

Case I	Feature	Description	Case II
Y	salary	2-salary intervals: $> 50K$ and $\leq 50K$	Y
S	gender	2 classes: male and female	S
X	relationship	6 classes of family relationships (e.g., wife and husband)	
	age	9-age intervals: $18 - 25$, $25 - 30$, $30 - 35$, ...,$60 - 65$	X
	workclass	8 types of employer + unknown	
	education	16 levels of the highest achieved education	
	marital-status	7 classes of marital status	
	occupation	14 types of occupation + unknown	
	race	5 classes	
	native-country	41 countries of origin + unknown	
	capital-gain	Recorded capital gain; (continuous)	
	capital-loss	Recorded capital loss; (continuous)	
	hours-per-week	Worked hours per week; (continuous)	
	education-num	Numerical version of education; (continuous)	



(a) Gender

(b) Relationship

(c) (Gender, Relationship)

Figure 6. Tradeoff between classification accuracy of non-sensitive feature (salary) and sensitive features (gender and/or relationship) under Case II for the UCI Adult dataset. Note that we use the classification accuracy obtained from the original testing dataset as the baseline performance, which is denoted by the green and red lines for the target variable (salary) and the sensitive variable (gender or/and relationship), respectively. In every plot, each point corresponds to a specific value of achieved test distortion (over all features except gender and relationship) ranging over $(0.58, 2.1)$, with distortion decreasing from the left to the right for each plot.

encoding to the categorical data. We restrict the continuous features in X to the interval $(0, 1)$ using normalization.

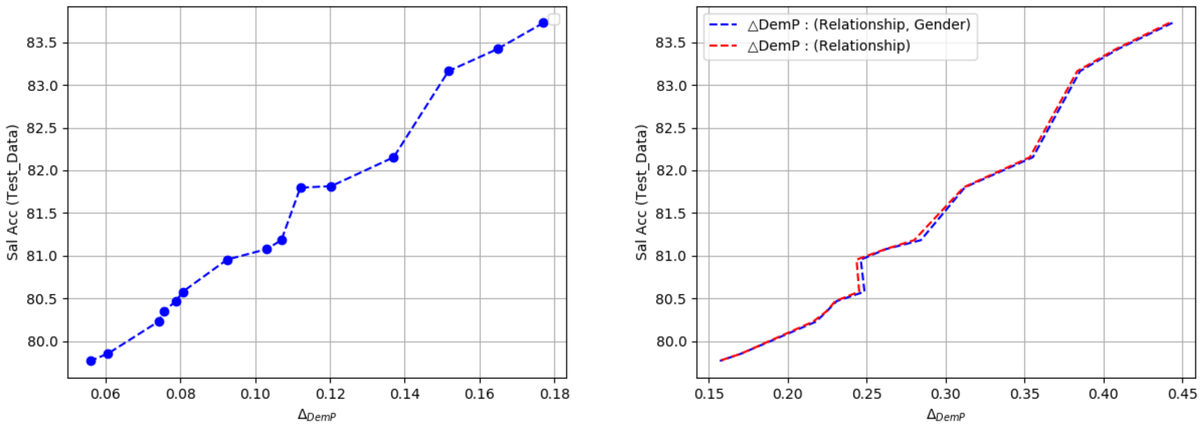
C.1.1. Additional Results

Case II: Non-binary Sensitive Feature.

Figures 6 and 7 illustrate the censoring and fairness performances of the generated X_r in hiding 'gender' and 'relationship', respectively, while preserving 'salary' information. Figure 6 illustrates the tradeoff between salary and sensitive feature S accuracies when S is either gender, or relationship, or both. From Figure 6, we observe that while the salary accuracy is above 79%, the classification accuracies of gender and/or relationship are about 66% (Figure 6a), 45% (Figure 6b) and 41% (Figure 6c), respectively. Note that the probabilities of male, husband, and the combination (male, husband) are 66%, 40% and 40%, respectively, in the original test data. Therefore, while the salary accuracy is preserved at 79%, the inferences of gender, relationship, and combination (gender, relationship) approach random guessing with these priors. Thus, our FUR framework can effectively hide one or more sensitive features. However, suppressing multiple correlated sensitive features comes at a cost of a reduction in salary accuracy. Thus, comparing Figures 2a and 6a, we see a maximal reduction of 3% in salary accuracy for a given gender accuracy[6].

For Case II, Figures 7a and 7b illustrate the tradeoffs between the salary accuracy and Δ_{DemP} for S chosen as gender or

[6] In Figures 2a and 6a, the baseline performances are different because for Case II, the feature variable X does not contain 'relationship'.



(a) Gender

(b) Relationship (or) (Relationship, Gender)

Figure 7. Case II for UCI Adult: Tradeoffs between salary accuracy and the Δ_{DemP} of gender and/or relationship. For the original test data, Δ_{DemP} for gender, relationship and the pair (gender, relationship) is 0.2, 0.438 and 0.443, respectively. In every plot, each point corresponds to a specific value of achieved test distortion (over all features except gender and relationship) ranging over $(0.58, 2.1)$, with distortion decreasing from the left to the right for each plot.

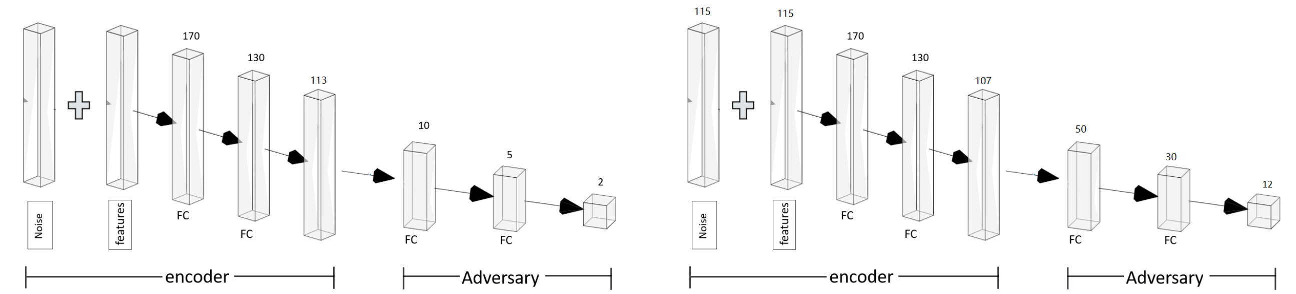
relationship or both. We observe that while salary accuracy is above 94% of the baseline performance, the value of Δ_{DemP} is dropped to 25% for gender and to about 34% for both relationship and their combination. In short, X_r works well in decorrelating gender and relationship both separately and jointly without affecting downstream classifier performance. From Figure 7b, we observe that the value of Δ_{DemP} for the combination is almost the same as that for relationship. In addition, comparing the results in Figures 2b and 7a, for any given Δ_{DemP} for gender, the salary accuracy in Case II is about 1% lower than that in Case I; this can be viewed as the cost of eliminating a potentially sensitive feature (relationship) that is also correlated with the target feature (see also, footnote 6). Finally, comparing the results in Figures 2b and 7b, for any given salary accuracy, Δ_{DemP} for gender in Case II is about 0.25 higher than that in Case I; this can be viewed as the effect of using non-binary sensitive features on Δ_{DemP}, now defined as the maximum over all values taken by the non-binary sensitive feature.

C.1.2. Architecture

The two architectures used are shown in Figure 8. We concatenate the pre-processed data with a same-size standard Gaussian random vector and feed the entire vector to the encoder. The encoder consists of two fully-connected (FC) hidden layers, the first with 170 neurons and the second with 130 neurons. Since the output representation X_r has the same dimension as the feature variable X, the output layer of the encoder has 113 (as shown in Figure 8a) and 107 (as shown in Figure 8b) neurons for Case I and Case II, respectively. We use a ReLU activation function in the encoder.

We recall the two cases considered for this dataset: Case I with binary sensitive feature S (gender) and Case II with non-binary S (gender and relationship). For Case I, the inputs can be either X only or both X and S. With only X as input to the encoder, the length of the input vector is 226, and when both X and S (binary) are inputs, the input vector length is 230. In both scenarios, the length of the encoder's output vector is 113. For Case II, since both X and S are inputs, the length of the input vector is 230. The length of the encoder's output vector is 107, since S is non-binary in this case.

For Case I, the adversarial classifier in Figure 8a consists of three fully-connected (FC) layers 1 to 3 with 10, 5 and 2 neurons, respectively, and it takes X_r as the input and outputs a probability distribution for the binary sensitive variable S (i.e., gender). Here, ReLU is used as the activation function in the two hidden layers and soft-max is used in the output layer to generate the probability distribution for gender. The same architecture is used for the downstream application of salary classification. For Case II, the adversarial classifier in Figure 8b consists of three fully-connected (FC) layers 1 to 3 with 50, 30 and 12 neurons, respectively. Leaky ReLU is used as the activation function in the two hidden layers and soft-max is used in the output layer. All of the above models use log-loss as the loss function and are optimized using Adam optimizer (Kingma & Ba, 2017).



(a) UCI Dataset: The architecture for Case I (S is the binary gender variable)

(b) UCI Dataset: The architecture for Case II (S is non-binary and includes both gender and relationship)

Figure 8. The architectures of the encoder and adversary for the UCI Adult dataset for both Cases I and II.

C.2. UTKFace Dataset Details

The UTKFace dataset (Zhang et al., 2017) consists of more than 20k 200×200 color images of faces labeled by age, ethnicity, and gender. Individuals in the dataset have ages from 0 to 116 years and are divided into 5 ethnicities: White, Black, Asian, s. Asian (Indian), and others including Hispanic, Latino and Middle Eastern. We take gender as S, the image as X, and age and ethnicity as two target labels Y. We reshape the original 200×200 color images (of faces) in the UTKFace dataset to color images of size 64×64 and we restrict the data to contain images for ages between 10 and 65.

C.2.1. Additional Results

Figure 9 illustrates the output X_r for 16 typical[7] faces in the UTKFace dataset for increasing per-pixel distortion. From Figure 9, we observe that: (i) for a small per-pixel distortion (e.g., 0.003), gender-distinguishing features such as lip color are smoothed out; and (ii) at higher per-pixel distortion (e.g., 0.006), the FUR framework can generate a face with an opposite gender (see the highlighted examples in Figure 9) thereby completely obfuscating this sensitive feature; (iii) when the average per-pixel distortion is too large (e.g., 0.01), the representations generated are often too blurred. Table 2 shows the decrease of Δ_{DemP} for each of the four ethnicities as the distortion increases.

For the downstream task of age regression, we use the mean absolute error (MAE), i.e., the average absolute difference between the predicted age and the true age, as the utility measure. Figure 10 shows the tradeoff between gender classification accuracy and the MAE for age regression. In Figure 10a, we observe that while the classification accuracy for gender is about 62%, which is a 35% decrease from the baseline performance of 94%, the increase in the MAE is 1.5 which is about a 20% increase from the baseline performance of 7.2 years. Figure 10b shows the cumulative distribution function (CDF) of the difference between the true and predicted age for various distortions, from which we can see that the drop of the cumulative probability is at most 1%. Thus, the generated FUR guarantees reliable performance for both age and ethnicity prediction; thus, constraining the distortion of the generated X_r can be effective in guaranteeing utility for multiple tasks.

In Figure 11, we illustrate the tradeoff between the MAE and Δ_{DemP} of the generated X_r in age regression. In Figure 11a, while preserving 86% of the utility baseline performance, the Δ_{DemP}, i.e., the maximal value of demographic parity measure over the 56 age values, decreases to 0.015, which is less than 33% of the $\Delta_{\text{DemP}} = 0.046$ in the original test data. Figure 11b shows the demographic measure $\Delta_{\text{DemP}}(y)$, $y \in [10, 65]$, for various distortions; we observe that when the pixel distortion is 0.01, even while $\Delta_{\text{DemP}} = 0.015$, $\Delta_{\text{DemP}}(y) = 0$ for 17 distinct ages. That is, the predictions of these 17 ages are completely independent of gender and DemP is achieved for those predictions.

C.2.2. Architecture

The architectures used are shown in Figures 12 to 14. Figure 12 illustrates the architecture of the FUR model, which consists of an encoder and an adversarial classifier. The encoder is implemented using a noisy autoencoder, whose encoder transforms the original 64×64 RGB-images into a 4096-dimensional feature vector. This feature vector is mixed with a

[7] The 16 typical faces covers the 8 possible combination of 2 genders (male and female) and 4 ethnicities (White, Black, Asian and Indian) and includes young, adult and old faces.

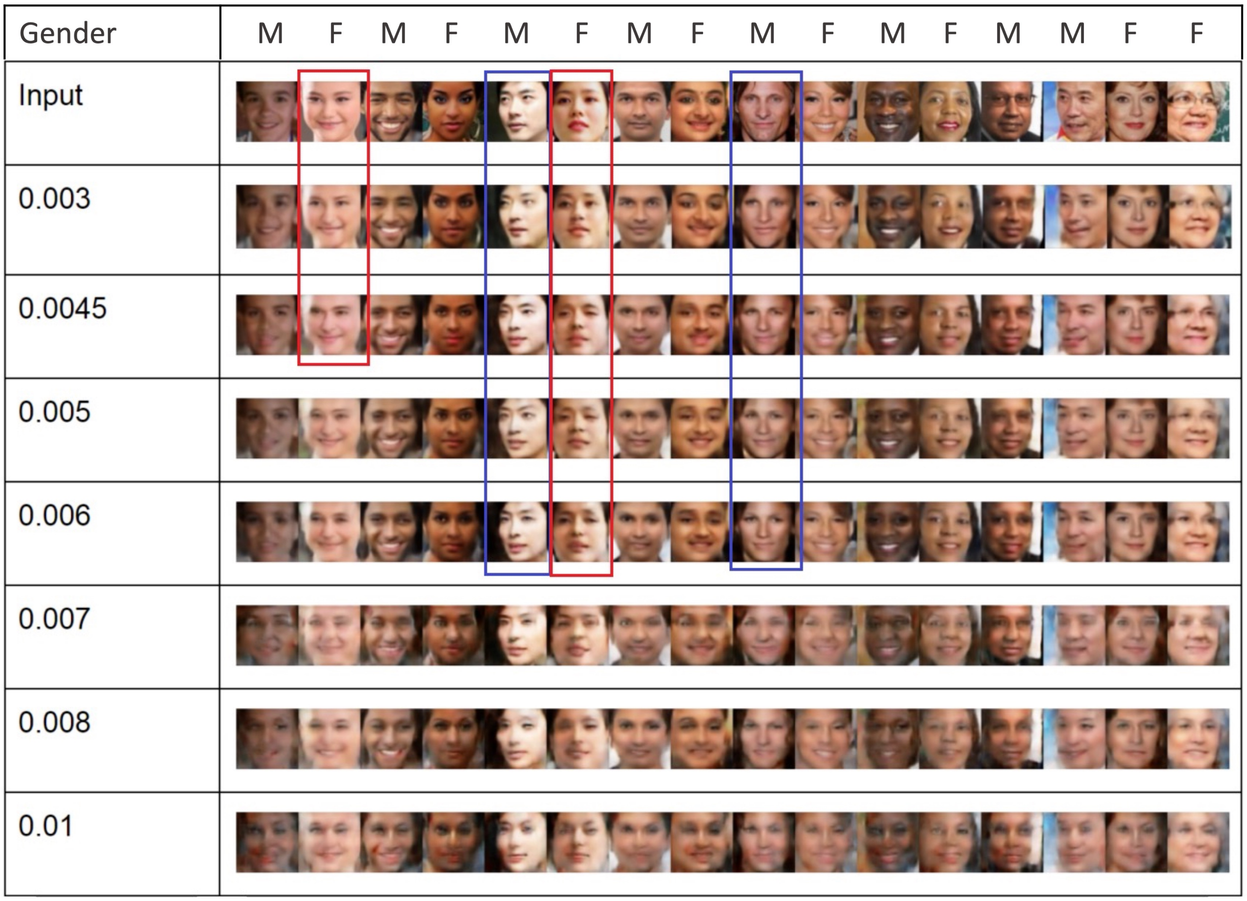


Figure 9. The encoded face images for different values of per-pixel distortions for the UTKFace dataset. Set of vertical faces highlighted in boxes makes explicit how the sensitive feature (gender) is changed with increasing distortion. The ground truth gender values for the images are shown in the top-most row.

4096-dimensional standard normal random vector[8] and then fed into a decoder to reconstruct a 64×64 colorful image, which is the universal representation X_r. This differs from a standard autoencoder, in which the feature vector is directly fed into the decoder. Specifically, the encoder part of the noisy autoencoder consists of four convolution layers 1 to 4 with 128, 64, 64 and 64 output channels, respectively, and three 2×2-max pooling layers following the first three convolution layers. The encoder part is followed by two fully-connected layers, each with 4096 neurons, which mix the noise and the output feature vector. The following decoder part consists of five convolution layers 1-5 with 64, 64, 64, 128 and 3 output channels, respectively, and three 2×2-up-sampling layers following the first three convolution layers. The adversarial classifier takes in the representation X_r and outputs the prediction of the sensitive S (gender). It consists of two convolution layers, the first with 20 output channels and the second with 40 output channels, two 2×2-max pooling layers following each of the convolution layers, and two fully-connected layers, the first with 40 neurons and the second with 2 neurons. The kernels in the convolution layers are of size 3×3. All convolution and fully-connected layers use ReLU as the activation function, except the last layers of the decoder and the adversary, which use sigmoid and softmax, respectively. The encoder and adversarial classifier use the square-loss and log-loss as the loss functions, respectively, and both are optimized using the Adam optimizer.

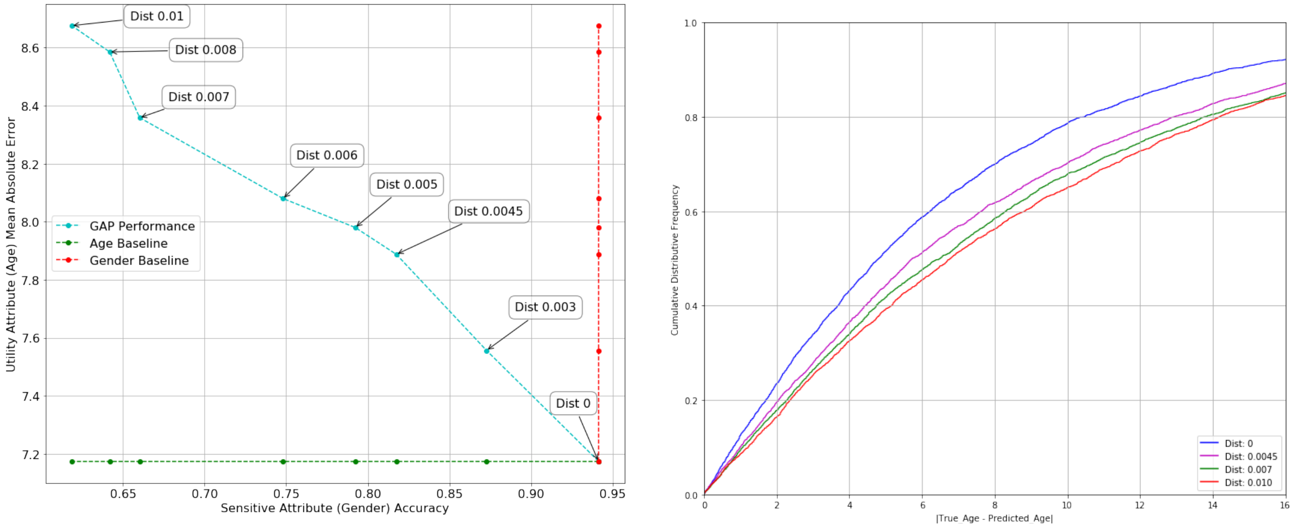
Figure 13 illustrates the architecture of the downstream non-binary classifier for ethnicity. The classifier is built by changing the top (last) 3 fully-connected layers of the VGG 16 model[9] pre-trained on ImageNet. The first layer has 256 neurons with

[8] A random vector is a standard normal random vector if all of its components are independent and identically distributed following the standard normal distribution.

[9] https://keras.io/applications/#vgg16

Table 2. Demographic parity fairness (indicated by $\Delta_{\text{DemP}}(\cdot)$) of ethnicity classification on the UTKFace dataset.

Distortion	0	0.003	0.0045	0.005	0.006	0.007	0.008	0.01
$\Delta_{\text{DemP}}(\text{White})$	0.061	0.055	0.04	0.03	0.03	0.02	0.02	0.01
$\Delta_{\text{DemP}}(\text{Black})$	0.109	0.021	0.02	0.05	0.03	0.05	0.03	0.03
$\Delta_{\text{DemP}}(\text{Asian})$	0.14	0.082	0.07	0.07	0.06	0.07	0.06	0.03
$\Delta_{\text{DemP}}(\text{Indian})$	0.031	0.006	0.01	0	0.01	0	0.01	0.01



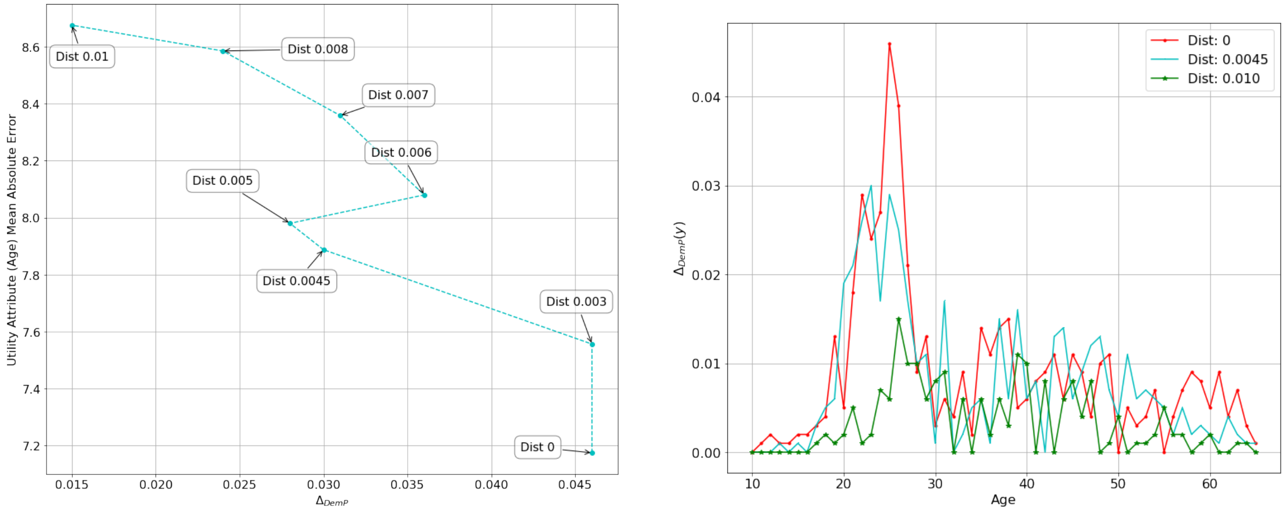
(a) Mean absolute error of age prediction vs. gender classification accuracy

(b) The CDF of the difference between the true and predicted age

Figure 10. Utility of age regression on the UTKFace dataset. Note that 'dist' indicates the per pixel distortion.

ReLU as the activation function and is followed by a dropout layer with the rate 0.5; the second layer has 4 neurons with softmax as the activation function. The classifier uses log-loss and is optimized by a stochastic gradient descent optimizer.

Figure 14 illustrates the architecture of the neural network used in the downstream application of age regression. The neural network consists of three 3×3 convolution layers 1 to 3 with 128, 64 and 32 output channels, respectively, three 2×2-max pooling layers following each of the convolution layers, and three fully-connected layers 1 to 3 with 512, 128 and 1 neurons, respectively. All layers use ReLU as the activation function, except the last layer, which uses linear activation. The model uses the squared loss as the loss function and is optimized using Adam optimizer.



(a) Mean absolute error of age prediction vs. Δ_{DemP}

(b) The demographic parity measure for various distortions

Figure 11. Achieved demographic parity for the age regression task on the UTKFace dataset. Note that in Fig. 11a, the x-axis is the maximal value of DemP in (11) over the chosen age range (10-65) and 'dist' indicates the per pixel distortion.

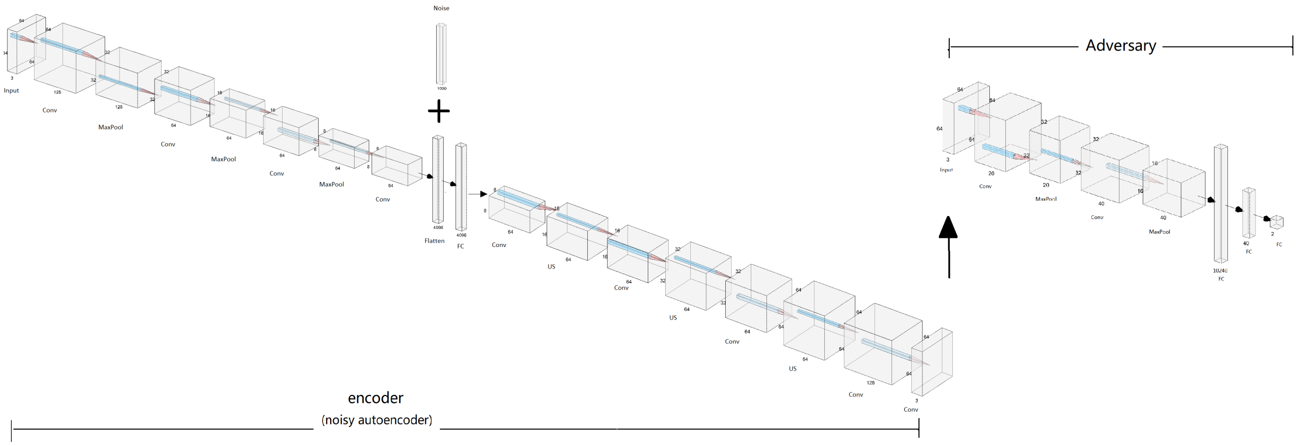


Figure 12. The architectures of the encoder and adversary for the UTKFace dataset.

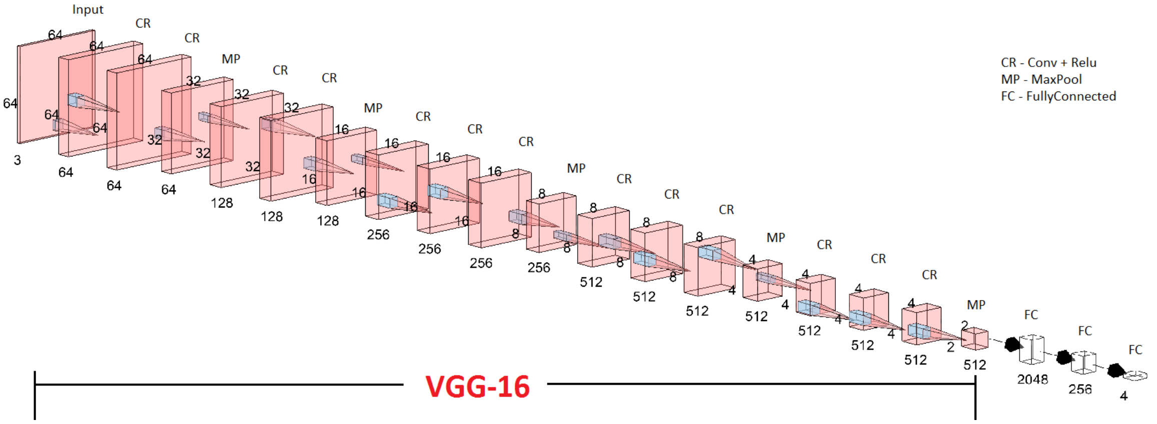


Figure 13. The architecture of the neural network for ethnicity classification for the UTKFace dataset.

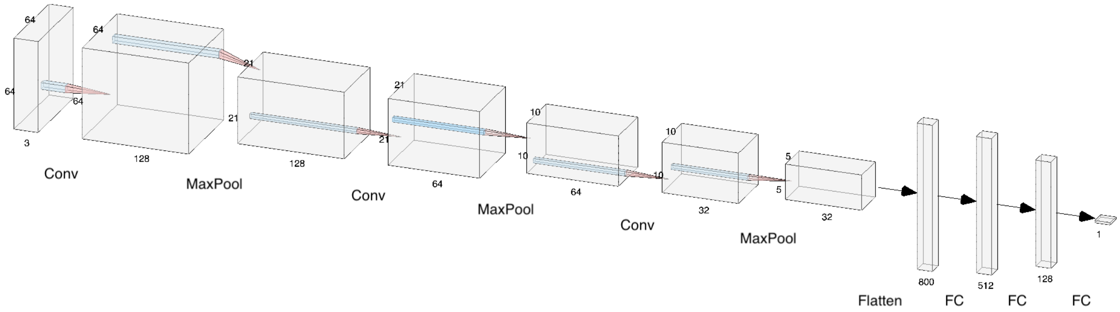


Figure 14. The architecture of the neural network for age regression for the UTKFace dataset.