CS 335 Project Report

# Translating from Unfair to Fair Embeddings

**Thomas Dimson**
tdimson@cs.stanford.edu

## 1 Background

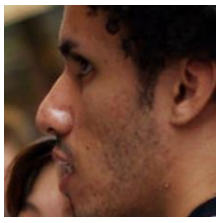
Machine learning researchers often think of images as *continuous data* and language as *discrete data* with tokens representing lookup indices into an embedding table. However, in the social media industry, images are often user data associated with an embedding at upload time (e.g. from a pre-trained vision network). Data pipelines treat those images as discrete data with $PhotoId$ acting as a lookup index. Like in natural language processing, downstream tasks do not require knowledge of how an embedding was generated.

That brings us to the question of **fairness**. Without knowing the mechanics of the original embedding computation, how can can we ensure downstream tasks do not implicitly encoding sensitive information? Our project examines the use of another neural network, dubbed the **translator network**, to produce an alternative embedding for a given photo. The translator network assumes nothing about the structure or training procedure for the original embeddings.

Below, in Section 2 we discuss some properties and challenges of choosing a dataset for training and evaluation. Section 3 illustrates the training procedure while treating race as a protected feature. Finally, in Section 4, we show success in translating embeddings through adversarial learning but caveat that training is unstable and translated embeddings lose some generality.

## 2 Dataset

(a) Annotated as female, middle eastern race and age 10-19



(b) Annotated as male, latino hispanic race and age 20-29

(c) Annotated as female, black race and age 20-29

(d) Annotated as male, white race, and age 40-59

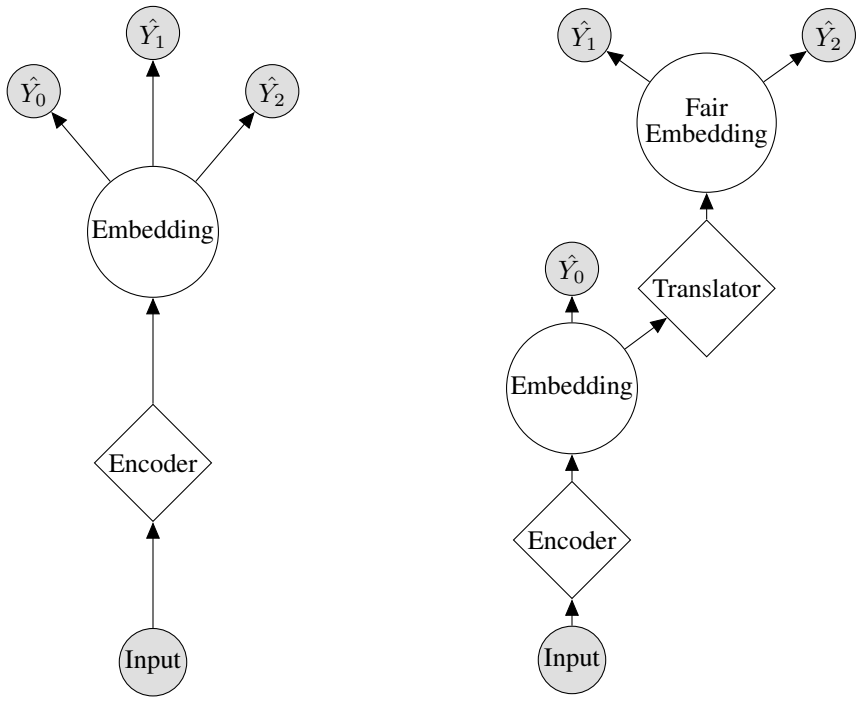
Figure 1: Samples from the Fairface dataset. Each example contains a square-cropped face image annotated with a gender (2 classes), race (7 classes) and age (9 classes, non-uniformly distributed) labels. Examples lack a uniform pose and are from different eras of photography.

Large scale vision datasets that are annotated with protected information are few and far between. At the project milestone we used the IMDB-Wiki dataset introduced in Rothe et al. [2018]; however in this final report use the FairFace dataset introduced in Kärkkäinen and Joo [2019]. FairFace, unlike IMDB-Wiki, is balanced across race and contains images annotated with three potentially protected attributes: race, gender and age cohort. The switch was motivated by the race annotation and greater data cleanliness.

FairFace is pre-split into training (86,744 examples) and validation sets (10,954). Gender classes are roughly balanced (47% female) as are the seven race categories (with Caucasians being over-

represented at 20%). Age labels are not balanced and are categorically annotated into nine different categories. Figure 1 shows a few examples from the dataset. In this project, race is treated as a protected attribute while making a gender prediction.

## 3 Approach




(a) Standard neural network encoder architecture

(b) Using our translation network

Figure 2: Contrasting a translation network with a standard neural network encoder architecture. The translator network is trained on top of an existing network and the reduces the ability for new networks to reconstruct protected labels.

The overall goal is to preserve as much of the original embedding as possible while removing any encoding of protected attributes. As shown in Figure 2, the components of the set-up are:

1. Given an image, an **encoder network** predicts an embedding for the image. The encoder network is frozen after its training with no further gradient updates applied.
2. Embeddings produced by the encoder become the training set for a **translator network**. The translator network consumes embeddings and produces fair embeddings while preserving shape. In other words, it is an auto-encoder.
3. Two heads are attached to the translator network: one that predicts a **control label** and one that predicts a **protected label**. In our dataset, the control label is gender and the protected label is race.
4. Using those heads, we train the translator network with a **fairness-regularizing loss** that tries to ensure it maintains control task performance while making *any other network* perform poorly at protected label prediction.

For our primary encoder network we use ResNet-34 as defined in He et al. [2015]. We experiment with two approaches: using pre-trained weights as-is as well as fine-tuning the network on our dataset to predict a multi-class $(gender, race)$ label. As described in Section 4.1, the latter approach appears to perform better.

### 3.1 Loss Function

The goal of translator is three fold: (1) to preserve the ability to make predictions on a control task, (2) to remove the ability to predict protected features and (3) to preserve as much of the structure of the original embeddings as possible. The following loss matches that structure:

$$Loss_{translator} = Loss_{control} + \alpha Loss_{fairness} + \beta Loss_{reconstruction}$$

The natural choice of $Loss_{control}$ is a softmax-loss which encapsulates the ability of the network to predict its control label.

The natural choice of $Loss_{reconstruction}$ comes from viewing it as auto-encoder for embeddings. We chose a Euclidean $Loss_{reconstruction} = ||embed_{fair} - embed_{unfair}||_2^2$ but any distance function would suffice.

The choice of $Loss_{fairness}$ is more interesting; we examine three choices from the literature:

- $Loss_{fairness} = -CrossEntropy(protectedLabel)$ attaching a new head on the embeddings to predict the protected label; Wadsworth et al. [2018] explored this loss to reduce bias in recidivism-prediction.
- $Loss_{fairness} = CrossEntropy(protectedLabel)$ modified so that it has a negative gradient during back-propagation; Raff and Sylvester [2018] successfully explored this loss as a general method for reducing discrimination.
- $Loss_{fairness}$ as an adversarial $CrossEntropy$ loss over another network that predicts the protected label. We formulate optimization as a minimax problem with alternating rounds of minimizing $Loss_{control} + \alpha Loss_{fairness} + \beta Loss_{reconstruction}$ with respect to the translator and maximizing $Loss_{fairness}$ with respect to adversary. Madras et al. [2018] explored this loss as a general framework to fairness.

Results using these loss functions are described later in Section 4.2.

### 3.2 Evaluation and Training Procedure

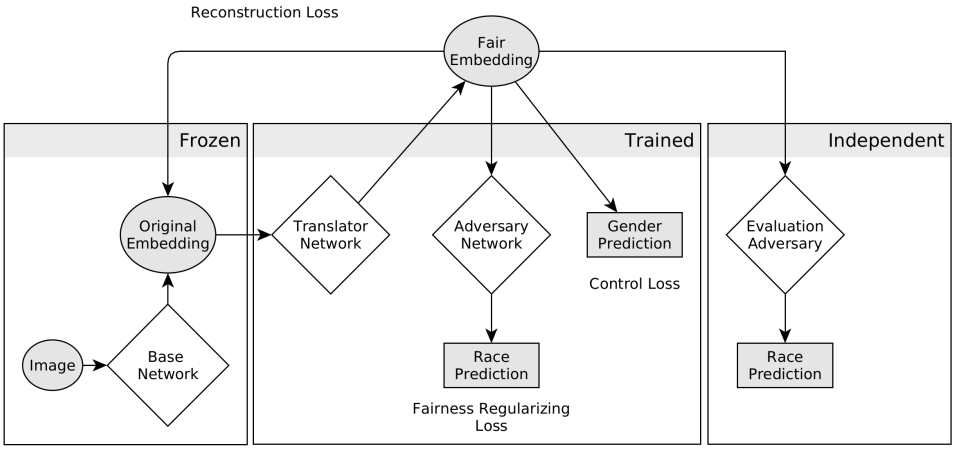



Figure 3: Information flow between the four networks and three loss functions involved in training and evaluation.

Unlike typical fairness problems, our goal is to remove protected information for *unknown* downstream tasks. We had originally examined the use of parity gap and opportunity gap (Beutel et al. [2017]) as a success metric. However, it is possible for the translator to show *equality of opportunity* while also encoding latent information about protected features. In other words, it appeared the fairness regularizers made the embeddings fair only relative to the instantiation of the control head.

We therefore evaluate success by the performance of an independent **evaluation adversary** trained to predict the protected label. This network is strictly used for evaluation and is trained independently from the original fairness network (which itself may be trained with its own adversarial loss).

As Figure 3 shows, the training procedure requires up to 4 networks:

1. the **base network** that to produce (unfair) embeddings
2. a **translator network** to produce fair embeddings with a control task head
3. depending on the loss, either an **adversarial network** predicting protected features or a **protected head** to the translator network.
4. an independent **evaluation adversary** trained to directly predict the protected feature

The translator is successful if the evaluation adversary has little or no predictive power over protected attributes. We choose F1-score as the metric of predictive power due to simplicity and disallowing a trivial bias-only solution. We were burned a few times by the adversary trivially learning a bias term that maximizes accuracy on the most represented protected class.

For simplicity, our protected race label is binarized into Caucasian and non-Caucasian. During translator training, we periodically re-train an adversarial evaluator and report its accuracy and F1 scores. Networks are trained against the training set but we take care to only use metrics calculated against the validation set.

### 3.3 Hyper-parameters and Tuning

Our hyper-parameter space is quite large because of the number of networks involved in training. Table 5 shows a list of hyper-parameters for our best model utilizing an adversarial loss. We search over the parameter space using a random search procedure.

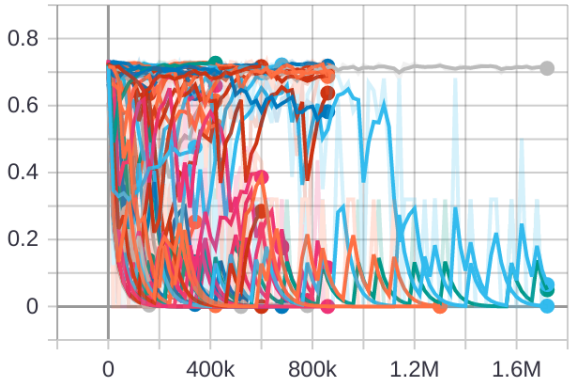



Figure 4: Various curves for the F1 score of the evaluation adversary predicting race. During translator training, we periodically re-train the adversary from scratch to predict race. Curves are unstable and vary intra-run and inter-run.

In practice, training outcomes were binary when using an adversarial loss: the control task and adversary either had full predictive power or none at all. Figure 4 illustrates the instability of outcomes in during training runs of the random search procedure. We choose a run that had full predictive power for the control task, no predictive power for the adversary and a high reconstruction weight; i.e. one that preserved as much of the original embeddings as possible.

## 4 Results

Much of this project was trying to get *something* working. We omit details of all our many failed experiments and are pleased to report that we eventually found a feasible network that conceals protected information. The network uses the hyper-parameters of Table 5 and an adversarial regularizing loss. It places a large (70%) weight on reconstruction of the original embeddings but removed race-specific information. Figure 6 contrasts nearest neighbors in the original embedding space

with those in the translated embedding space and shows a much greater racial diversity amongst candidates.

| Parameter | Best Run | Description |
|---|---|---|
| objective weight | 0.27059 | Blend weight to put on control loss |
| reconstruction weight | 0.71765 | Blend weight for reconstruction loss |
| adversary weight | 0.011765 | Blend weight for adversarial loss |
| adversary every | 10 | Minimax balance; how many minimization steps before applying adversary maximization |
| translator squeeze | 2-layer relu 512->60->512 | Structure of the translator |
| learning rate | 0.0088587 | Learning rate for translator network |
| num epochs | 4 | Number of training epochs |
| adversary lr | 0.0088587 | Learning weight for adversarial loss |
| weight decay | 0.27826 | L2 regularization for training / adversary |
| batch size | 0.008587 | Batch size for training |

Figure 5: The best hyper-parameters found during a random search procedure. In practice, we found that outcome was binary; either the adversary activated fully o not at all. We chose a variant that activated the adversary but also retained a high reconstruction weight.

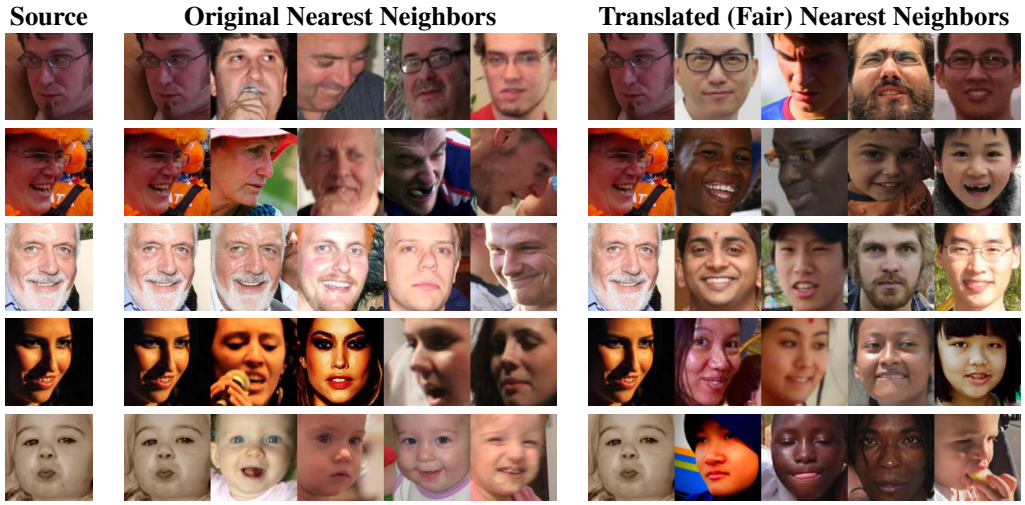



Figure 6: Euclidean nearest-neighbors for validation set examples in the two latent embedding spaces. The original nearest neighbors are cleary aligned along gender and race lines while the translated nearest neighbors retain gender information without race.

### 4.1 Baseline Vs. Model

We also examine the choice of **base network** for producing the underlying unfair embeddings. The goal is unfair embeddings containing linear predictive power over race and gender. We started with a ResNet-34 model and trained simple linear layers (Adam optimizer, lr=0.0001) on their embeddings with two approaches:

- No-fine tuning. Linear layer predicting gender=male achieves 0.761 F1-score on the validation set and predicting race=Caucasian achieves 0.346 F1-score on the validation set.
- Fine-tuning ResNet-34 first by a $(gender, race)$ categorical label (14 categories). Afterwards, a linear layer predicting gender=male achieves 0.928 F1-score on the validation set and predicting race=Caucasian achieves 0.734 F1-score on the validation set.

Results were relatively stable across choice of hyper-parameters. Since fine-tuning performed better we use it as our base model. Its associated metrics (0.928 F1 on gender=male and 0.734 F1 on race=caucasian) also become the **baseline** for our fairness translator. We seek a model that has

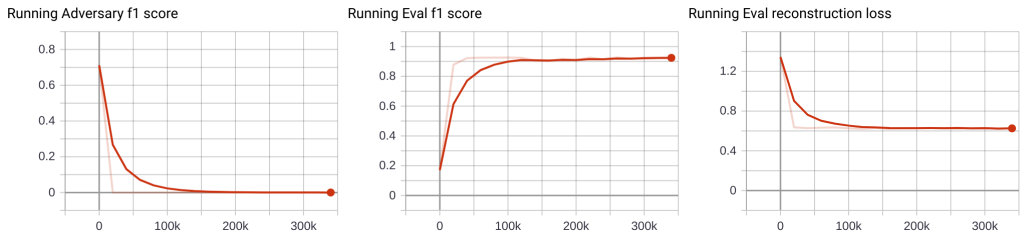



Figure 7: Learning curves during training for our best model run. With our choice of hyper-parameters we were quickly able to obtain validation-set performance of predicting gender (eval f1), while having no information about race (adversary f1) and retaining embedding semantics (reconstruction loss).

nearly the same gender prediction performance where an adversary predicting race=Caucasian has comparatively worse performance.

Our final model, with hyper-parameters shown in Table 6, achieves a F1 score for predicting gender=male of 0.9242 on the validation set, nearly matching our original baseline. Conversely, an independently trained linear adversary has no predictive power on race=Caucasian and achieves a F1 score of zero. The results for the independent adversary were also relatively stable across hyperparameters.

Although not our primary objective, our best model also shows a small demographic parity gap (0.01) and equality of opportunity gap (gender=male 0.01, gender=female 0.02) confirming that race information is not being used in gender prediction.

### 4.2 Alternative Losses

While our best model uses an adversarial loss, we also experimented with the other fairness regularizing losses previously mentioned in Section 3.1.

Unlike Wadsworth et al. [2018], training with $Loss_{fairness} = -CrossEntropy(protectedLabel)$ forced our loss function to diverge to negative infinity. Our network was free to learn an arbitrarily bad predictor which drove cross-entropy to negative infinity. We tried constraining this model by adding larger $L_2$ regularization but training remained unstable.

We also tried using $Loss_{fairness} = CrossEntropy(protectedLabel)$ but flipping the sign of the gradient during back propagation. After an extensive random search we found that the model struggled to maintain control task performance when minimizing the fairness loss. Reconstruction loss also oscillated during training which suggests the negative gradient may hamper the ability of the model to structure embeddings.

### 4.3 Alternative Models

With so many models in our task, there are arbitrarily large numbers of modeling choices that could have been made. During the course of training we tried different base models (e.g. ResNet18, ResNet54) and a plethora of adversarial regularizers (one layer, two layer and three layers). Most of these experiments either failed or had identical performance to our model above; we leave a full exploration of causes to future work.

### 4.4 Generalization of Embeddings

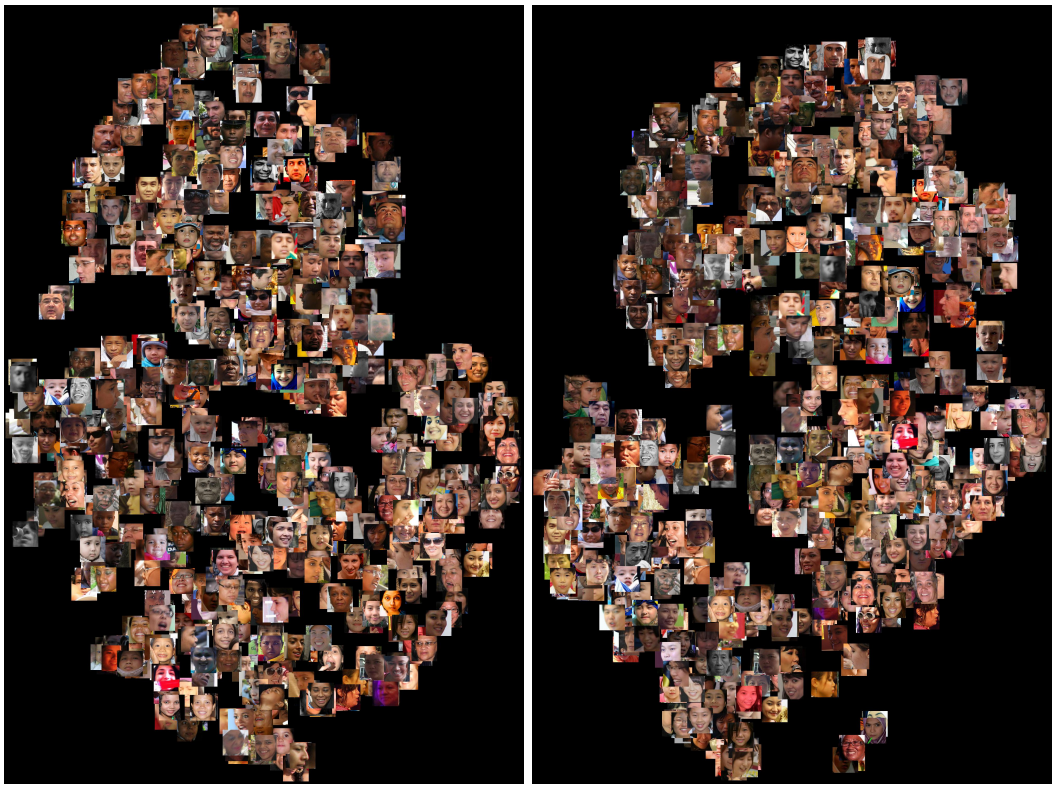
The re-construction loss helps ensure that our embeddings are as close as possible to the original embeddings. This raises the question: would a model trained on the original embedding maintain performance when evaluated using the fair embeddings?

We test this by using FairFace dataset's age label; binarizing into $age < 30$ (negative) and $age >= 30$ (positive). We first trained a linear model on the original embeddings that was able to achieve a F1-score of 0.70 against the validation set. We then evaluated with the *fair* embeddings against the validation set. During early stages of training it is able to generalize with a maximum eval F1-score of 0.55. As training continues, the network appears to specialize to the unfair embeddings and

fair embedding evaluation metrics diverge (down to zero). It is a promising result but leave deeper explorations to future work.

## 5 Conclusion

Above, we describe a method removing protected information from pre-trained embeddings without accessing the original network. We prove its efficacy by removing latent racial information encoded in a ResNet34 model fine-tuned on the FairFace dataset. Doing so required adversarial training that was unstable but effective. We also examined alternative losses but diverge or have poor predictive power at the control task.



(a) t-SNE reduction of fair embeddings

(b) t-SNE reduction of unfair embeddings

Figure 8: Visualization of our two embedding spaces, with unfair embeddings showing clear racial patterns.

At the onset of this project we were unsure whether our task was feasible. Most of our time was spent developing a model that did not diverge during training. As our project progressed, we eliminated non-adversarial methods and discovered that fairness regularizers targeted reduce prediction performance but don't necessarily remove latent information. Given more time, our next steps would be to explore more datasets, try to characterize *why* certain models can fairly translate, and explore different formulations of the reconstruction loss. Thank you!

## References


A. Beutel, J. Chen, Z. Zhao, and E. H. Chi. Data decisions and theoretical implications when adversarially learning fair representations. *CoRR*, abs/1707.00075, 2017. URL `http://arxiv.org/abs/1707.00075`.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

K. Kärkkäinen and J. Joo. Fairface: Face attribute dataset for balanced race, gender, and age, 2019.

D. Madras, E. Creager, T. Pitassi, and R. Zemel. Learning adversarially fair and transferable representations, 2018.

E. Raff and J. Sylvester. Gradient reversal against discrimination, 2018.

R. Rothe, R. Timofte, and L. V. Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2-4):144–157, 2018.

C. Wadsworth, F. Vera, and C. Piech. Achieving fairness through adversarial learning: an application to recidivism prediction, 2018.