Fake Face Detection

Alex Kyllo John Wyman Will Thomas

December 12, 2020

Abstract

In this study we investigate whether it is still feasible to automatically discern AI-generated human face images from genuine photographic ones, by training a convolutional neural network on a labeled dataset of 70,000 real and 70,000 fake face images. We use the fake face classification problem to further explore the topic of model fairness, by evaluating the model's performance across age, gender and race groups on a demographically labeled face dataset. To achieve this, we propose a method of utilizing an autoencoder network to translate demographically labeled real face images into an approximation of their latent space code representations and then reconstruct them back into images, creating a dataset of matching fake face images with the same demographic labels. This allows us to assess whether our fake face detection model works equally well for human faces of different age, gender and race groups, or whether it even generalizes to a dataset that is demographically diverse and balanced.

Introduction

Generative Adversarial Networks (GANs) have created the ability to encode photographic images into a latent space representation and automatically generate many images that can appear to be genuine photographs, to the human eye. NVIDIA's StyleGAN [1] model, trained on a dataset of human face images, is capable of generating extremely photorealistic images of people who do not exist. StyleGAN is an example of a generative adversarial network (GAN), a type of network that consists of two models, a *generator*, which learns a data distribution from a training set and draws from it, and a *discriminator*, which learns to estimate the probability that a sample came from the training data or from the generator [2].

The application of GANs to generating realistic human faces has garnered much media attention in recent years. The well-known website This Person Does Not Exist serves random, high-resolution fake face images drawn from StyleGAN. Many models have been created for exploring the latent space of StyleGAN to discover vectors that correspond to semantic directions in which the latent codes can be perturbed in order to make semantic edits to human face photos, such as making the face look older vs. younger, more masculine vs. more feminine [3], or thinner vs. heavier [4]. The popular mobile application FaceApp utilizes this method to allow users to transform images of themselves in this manner and share them on social media for humor. Two different people's face images can also be blended together by encoding both images and interpolating some point between them in the latent space, which can be used for applications like visualizing what a couple's children might look like, a concept that has been implemented in the project FamilyGAN.

It is also easy to imagine nefarious applications of fake face generation, some of which have already been realized, such as swapping faces of celebrities onto pornographic actors or falsifying speeches by political leaders [5]. Therefore, there is a public interest in retaining the ability to effectively discern real human face images and video from fake ones and detect "deep fake" tampering, which requires training more deep learning models to distinguish them.

A problem with the available open datasets of human faces used in training deep learning models, is bias in the demographic composition of the pictured individuals. The machine learning community has recently been struggling with the issue of model fairness–it is important that models perform equitably for users and data subjects of different backgrounds, and also very difficult to enumerate and quantify the sources of bias in training data that can contribute to biased model performance.

The FairFace [6] study introduced a new dataset of human face images collected from public datasets with manually verified, crowdsourced age, gender and race labels. The FairFace paper demonstrates that because existing public datasets of human faces contain a majority of white faces, models trained on them fail to generalize well to datasets where more non-white faces are present. We suspected that this might also be the case for the 70k real and fake faces dataset that we utilized for model training, and sought to test this by evaluating it on a demographically labeled dataset.

While the FairFace dataset provides real human face images that can be used to assess disparities in a fake face detector's true negative and false positive rates, a second, similarly labeled dataset of fake face images is needed to compute true positive and false negative rates for specific age, gender and race groups. To address this gap, we investigated methods for "falsifying" a real face image by autoencoding it via the StyleGAN latent space. A research team at Tel Aviv University recently developed a novel encoder network [7] that is capable of approximately reconstructing StyleGAN's latent code representation of a face image and then decoding it back into an image, leading to a fake face output image that very closely resembles the real face input image, implying that the original demographic labels would remain valid.

Methods

Data Preprocessing and Augmentation

We tested several methods for preprocessing and augmenting the image data before feeding it into the CNN model.

- 3-color (RGB) images vs. grayscale
- Pre-cropping and centering faces using pre-trained face detection models
- Random horizontal flips

For pre-cropping, we utilized two different pre-trained face detection models, MTCNN and Dlib. (TODO: citations)

Model Training

To solve the binary classification task of distinguishing between real and fake human face images, we trained several variations of deep Convolutional Neural Networks (CNN), varying the number of convolution layers as well as several model hyperparameters and image preprocessing steps.

Our intial baseline model was a CNN with three convolution layers using a 3x3 element kernel.

TODO: learning curves for baseline and best model

Model Serving

TODO: Details and screenshots of web application here

Model Explainability

TODO: Explanation and screenshot of eli5 highlighted image, possibly CNN activation map

Model Evaluation

Our primary metric for performance assessment during training and model selection was validation set accuracy, because the balanced classes of the input dataset made accuracy straightforward to interpret. For final model performance on out-of-sample test data, we break down performance with a 2x2 confusion matrix and report F1 score, precision score and recall score in addition to accuracy score.

For fairness metrics, we selected two standard binary classifier metrics to evaluate our classifier on: disparate impact ratio, given by the ratio of the rate of positive predictions for the unprivileged class to that of the privileged class [8], given by:

$$\frac{P(\hat{y}|unprivileged)}{P(\hat{y}|privileged)}$$

and average odds difference, given by:

$$\frac{(FPR_{unpriv} - FPR_{priv}) + (TPR_{unpriv} - TPR_{priv})}{2}$$

for the following binary group definitions taken from the FairFace labels:

1. Gender = "male" vs Gender = "female"
2. Race = "white" vs all other races
3. Race = "black" vs all other races
4. Age = "0-2" vs all other ages
5. Age = "3-9" vs all other ages
6. Age = "more than 70" vs all other ages

We examine the model fairness for children and senior citizens as a recent study [9] found that the popular face recognition model Face++ disproportionately fails to recognize children's faces in images collected from social media.

Results

We determined that the fake face classification task is still achievable with a relatively simple CNN model, but that model also failed to generalize to the unseen FairFace dataset.

Preprocessing Results

We utilized two pre-trained face recognition models to locate the human face in the image, align it so that the eyes, nose and mouth are level and centered, and crop to a margin around the face. Because these preprocessing models are themselves probabilistic machine learning models, they sometimes fail to recognize a human face at all (false negative) or incorrectly recognize some other object as a human face (false positive). We examined the images for which face detection failed.

The Dlib frontal face detector method mistook several objects including a logo and a necklace for human faces (Figure 1), while the MTCNN face detector method failed to identify several faces with brightly colored hair or wigs and heavy makeup as faces (Figure 2)

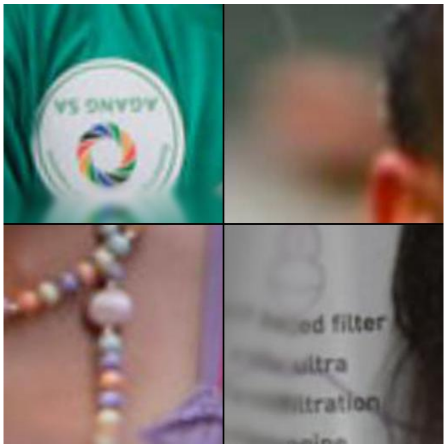

Figure 1: Sample of false positives cropped by Dlib.

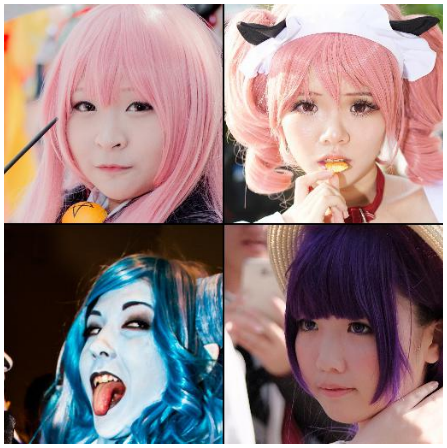

Figure 2: Sample of false negatives that MTCNN failed to crop.

By applying the pixel2style2pixel encoding algorithm we were able to successfully obtain a dataset of 79,484 matching pairs of real and fake

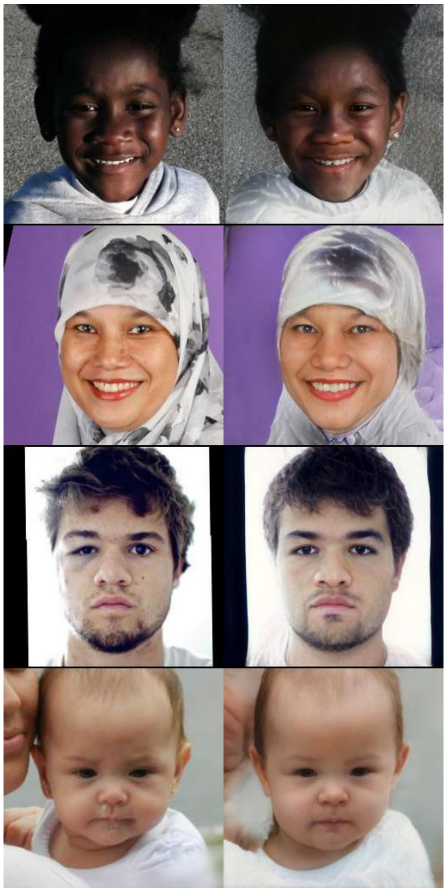

Figure 3: Selected FairFace images before/after encoding by pixel2style2pixel

Model Performance Results

The baseline model achieved the following performance on the Fake Faces test set, after 18 training epochs (Table 1):

Table 1: Baseline model performance

Accuracy	F1	Precision	Recall
0.942197	0.941569	0.951865	0.931493

Our best model in hyperparameter tuning was a variant of the VGG architecture [10] with 10 total layers (8 convolution layers and 2 fully connected layers). This model's performance is given by (Figure 2).

Table 2: Model performance on Fake Faces test set

Accuracy	F1	Precision	Recall
0.968298	0.967689	0.986595	0.949495

Fairness Assessment Results

Our best model trained on the 70k real and fake faces dataset failed to generalize to the FairFace dataset, marking nearly all of the FairFace observations as fake faces, resulting in high recall but very low precision and accuracy scores (Figure 3).

Table 3: Model performance metrics

Accuracy	F1	Precision	Recall
0.54305	0.676667	0.52357	0.9563

We originally intended to perform a fairness assessment on both our baseline model and our best model trained on the fake faces dataset, but because these models completely failed to generalize to the FairFace data, we decided not to even assess their demographic fairness metrics. Rather, we retrained our best model on a combined dataset consisting of the entire training sets from both the 70k real and fake faces and the FairFace datasets.

The resulting model performed very well on the combined dataset (Figure 4), and even performed better on the 70 real and fake faces dataset than the model version that was only trained on that dataset (Figure 5), suggesting that the additional training examples from FairFace led to a general improvement.

Table 4: Model performance metrics on FairFace + Fake Faces combined dataset

Accuracy	F1	Precision	Recall
0.984457	0.984411	0.986268	0.982561

Table 5: Performance on Fake Faces test set for model trained on combined dataset

Accuracy	F1	Precision	Recall
0.971232	0.971092	0.97374	0.968458

Our best model, using the VGG10 architecture and trained on the combined dataset, achieved excellent performance by the fairness metrics, disparate impact ratio (where 1.0 is perfectly fair) and average odds difference (where 0.0 is perfectly fair).

Table 6: Model Fairness metrics

	Disparate Impact	Avg Odds Diff
Male	0.993955	-0.000031
White	1.027800	-0.001012
Non-Black	1.004817	0.000983
Non-Child	1.011344	0.000167
Non-Senior	1.003951	0.001968

Discussion

Our initial goal of detecting fake faces was quite successful, especially on the original dataset. The <Insert finalized model name> managed an accuarcy of <x> percent when testing with the initial dataset. After some thorough testing and investigation, we made the decision to incorporate other fake images, and to go as far as making it more fair.

After a successfully testing our network against same set testing images, we thought that our model was doing quite well so we moved onto external datasets. Upon testing with other fake images, we noted that the accuracy took a dramatic turn in the opposite direction. While investigating, we determined that there could be several causes for this decrease:

1. The original training set had a specific pattern the our models were detecting, causing our network to look for those specific details. It could be something minute, such as the eyes

having x and y coordinates that were within a specific range. Although, we deteremined that this was unlikely due to our randomness when pre-processing (add noise, flips, shears, rotations) our data.

2. Our method for creating a fair face dataset was flawed <talk about how we got this dataset>
3. We didn't have enough data in original dataset to allow for a higher classification rate outside of that set.

Future Work

Overall, we are quite happy with the way our model classifies, but there are several adjustments we'd love to make that could help increase accuracy. We do not believe that there were any fundamental flaws in our methods, rather an inadequate amount of time, resources, and overall data. All of issues revolve around a lack of time, and given more, we could increase the performance of our model architecture and classification.

Time was our largest blocker throughout the entirety of our project, whether it was the time required to set up CUDA, train, or research new theories. During our models training time, we never saw a hint at over fitting, which indicated that there was still performance to be had. Given more time and compute power, we could have increased our accuracy. In addition, we could also take some images from NVIDA's fake face dataset and add that to our training data, but once again the cost involved is more than our team could handle.

In the future, we would love to incorporate these changes and throw more compute power at the problem. Using ample amounts of training data and more computer vision techniques we believe that there is still room to improve on our model. Although, as fake faces become better in better, we wonder how our model would fare. Looking at a company like NVIDIA, who has the resources to make fake faces using GANNs and enormous amounts of compute power, will it be possible to detect their fake faces?

References

[1] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[3] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing, 2020.

[4] V N S Rama Krishna Pinnimty, Matt Zhao, Palakorn Achananuparp, and Ee-Peng Lim. Transforming facial weight of real images by editing latent space of stylegan, 2020.

[5] Thanh Thi Nguyen, Cuong M. Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, and Saeid Nahavandi. Deep learning for deepfakes creation and detection: A survey, 2020.

[6] Kimmo Kärkkäinen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age. *arXiv preprint arXiv:1908.04913*, 2019.

[7] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. *arXiv preprint arXiv:2008.00951*, 2020.

[8] Christine Allen, Carly Ahmad, Muhammad Eckert, Juhua Hu, and Vikas Kumar. fairMLHealth: Tools and tutorials for evaluation of fairness and bias in healthcare applications of machine learning models. https://github.com/KenSciResearch/fairMLHealth, 2020.

[9] A. Mashhadi, S. G. Winder, E. H. Lia, and S. A. Wood. Quantifying biases in social media analysis of recreation in urban parks. In *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–7, March 2020.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.