Mid Project Report: Fake Face Detection

John Wyman, Alex Kyllo, Will Thomas

2020-11-25

Abstract

Our machine learning project is to develop a binary classifier model that can discriminate between real human face images and fake face images as generated by StyleGAN, a generative adversarial network (GAN) model developed by NVIDIA. Our secondary objectives are to deploy our model as part of an explainable inference application and to conduct a model fairness assessment that indicates whether the model performs equally well across age, gender and race groups.

Details

Our initial model development efforts yielded the finding that a Convolutional Neural Network (CNN) is capable of learning the fake face classification task. Our baseline model, a CNN with three convolution layers and dropout normalization, achieved 86.7% validation accuracy after 50 epochs. Our best model so far, a VGG-style architecture with 8 convolution layers, 2 fully connected layers, batch normalization and ADAM optimization, achieved 96.7% validation accuracy after training for 50 epochs on a single RTX 2060 Super GPU. We are exploring several other architectures such as VGG16, Xception, DenseNet, ResNet and EfficientNet.

When dealing with images, we realized that it would be computationally expensive to iterate over each image when training, given the high dimensionality in the input image sizes of 256 x 256 x 3. We decided to try and reduce the complexity of training by reducing the number of pixels in each image. The following steps were taken to reduce cost or training our models:

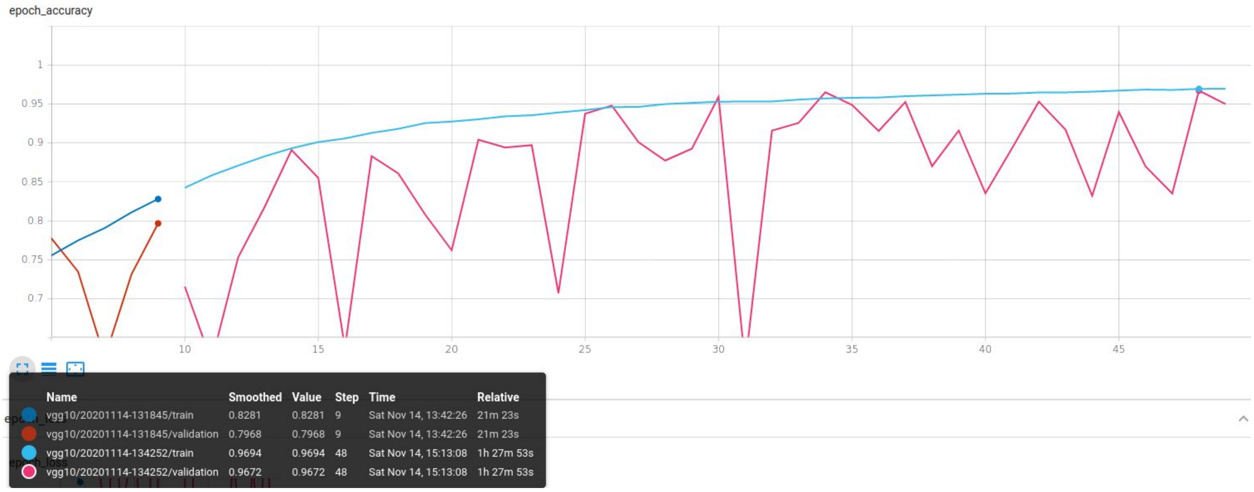
1. We utilized a pretrained MTCNN face detection algorithm to identify faces and crop them accordingly.
2. We leveled and centered the images around the face.
3. We converted the input images to grayscale to remove the color channel dimension.
4. We resized the input images to 128 x 128 pixels.

Upon applying our computer vision techniques, we saw a slight bump in both performance and training time. When we trained our baseline model using cropped and uncropped images, the cropped model outperformed the uncropped model by a 4% uplift in validation accuracy at the same number of epochs.

Our results up to this point have been promising, seeing an accuracy of 96.7% on the validation set. When training past 50 epochs, we are noticing severely diminishing returns, so we might gain more benefit from tuning rather than continuing training. At a later point, we will try reworking our network to try and achieve a higher accuracy. We suspect that there are still some accuracy gains to be had, but they will likely rely more on data pre-processing and hyperparameter tuning than on changing our network architecture or overall approach to the problem.

Data Analysis

Our best model so far achieved 96.7% accuracy at epoch 48, but validation set accuracy over time appears noisy, which raises concern about model generalization. We may experiment with other values for hyperparameters such as learning rate, normalization momentum, minibatch size, or choice of optimizer algorithm, to attempt to achieve smoother validation accuracy growth.



	Name	Smoothed	Value	Step	Time	Relative
●	vgg10/20201114-131845/train	0.8281	0.8281	9	Sat Nov 14, 13:42:26	21m 23s
●	vgg10/20201114-131845/validation	0.7968	0.7968	9	Sat Nov 14, 13:42:26	21m 23s
●	vgg10/20201114-134252/train	0.9694	0.9694	48	Sat Nov 14, 15:13:08	1h 27m 53s
●	vgg10/20201114-134252/validation	0.9672	0.9672	48	Sat Nov 14, 15:13:08	1h 27m 53s

Figure 1: Learning curve for our best model over 50 epochs

An analysis of the performance of our best model on the validation set reveals that most of the prediction errors were false negatives, giving us a higher precision than recall.

		Actual class	
		fake	real
Predicted class	fake	9477	137
	real	519	9863

Table 1: Confusion matrix for our best model on the validation set

Based on the validation confusion matrix (Table 1), we computed standard binary classifier performance metrics as follows:

Accuracy: 0.967
Precision: 0.986
Recall: 0.948
F1 Score: 0.967

Challenges

The remaining challenges we are facing are primarily in the areas of image preprocessing and model tuning. The search space of potential preprocessing functions and their parameters, as well as CNN model architectures and their hyperparameters, is unbounded. In order to maximize our chances of improving upon our initial results, we will need to develop a more systematic approach to experimentation that enables us to vary specific hyperparameters and save clearly identifiable experiment results for direct comparison. If we can find or design a simple tool for running isolated, reproducible experiments with a saved set of hyperparameters, we will be able to iterate on model training more rapidly and achieve a better final model.

Next Steps

Our remaining work in the project consists of three tasks: building a simple web application for explainable inference, conducting a fairness assessment for model performance, and further tuning of the CNN architecture, hyperparameters, and image preprocessing as time allows.

For the explainable inference application, we are developing a single-page website using Azure Functions and Vue.js, where a user can upload an image, and the model will process it and display a prediction value that is one of, “real face,” “fake face,” or “not a face.” It will also display a version of the original image with the pixels that were weighted most heavily in that prediction highlighted in a bright color, to help the user understand why the model arrived at its prediction.

For the fairness assessment portion of the project, we are utilizing a source of 108,501 demographically labeled real face images called the FairFace dataset and processing it through a pre-trained encoder network called Pixel2Style2Pixel, whose purpose is to encode input images into vectors in the latent space of the StyleGAN network. When these latent vectors are decoded back into images, they produce fake faces that resemble the real inputs. Below are examples of two demographically labeled real faces (left column) with their fake face equivalents (right column) reconstructed from latent vectors:

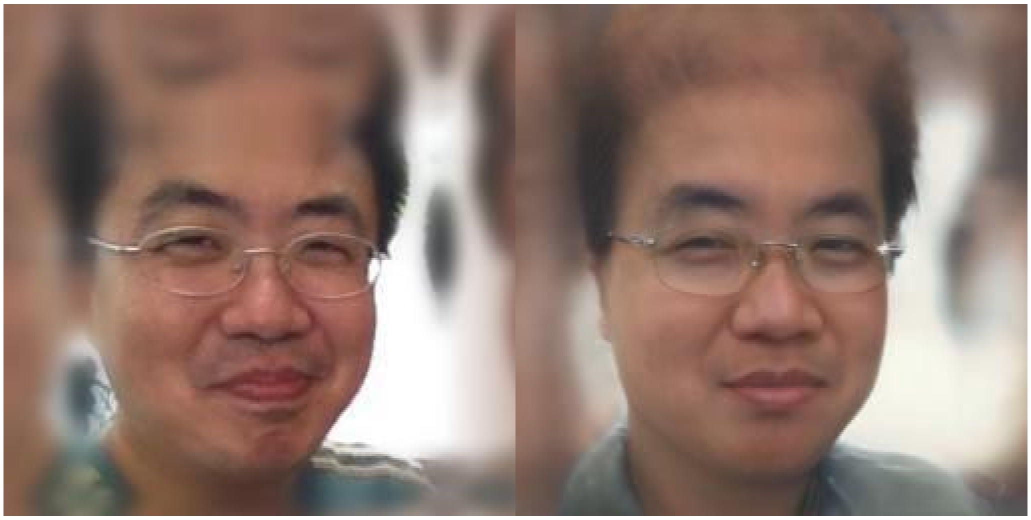


Figure 2: Matching real and fake faces, Age: 20-29, Gender: Male, Race: East Asian

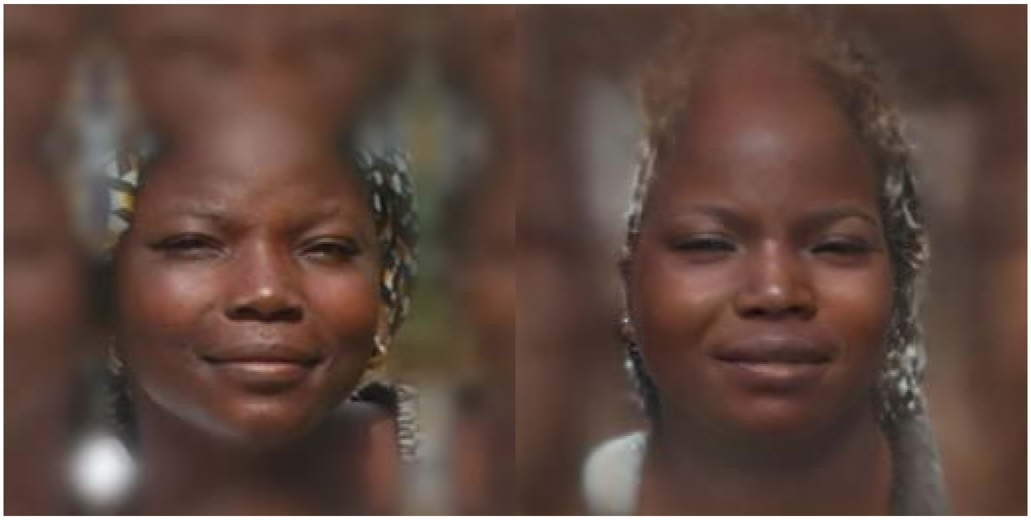


Figure 3: Matching real and fake faces, Age: 20-29, Gender: Female, Race: Black

By utilizing both the real and the corresponding fake images as a labeled test set for our model, we will be able to assess our fake face detection CNN model's performance across age, gender and race groups, comparing both false positive and false negative rates. Furthermore, if the model's performance is unsatisfactory for some demographic group, we can explore augmenting the original training and validation sets with examples from the FairFace dataset to improve its fairness.