Algorithmic Biases in Facial Recognition

Whitney Jablonski, Paul Jureidini, Samata Kurhekar, Shourya Veeraganti

Contents

Summary

Add in the summary when results section is finished.

1. Introduction

Facial analysis and facial recognition are technologies fueled by artificial intelligence that are becoming more entrenched in our every day lives even in a broad range of areas. Applications range from smartphone cameras (e.g., auto-focusing), law enforcement (e.g., border security, risk assessment for recidivism), and advertisement targeting (e.g., from profile pictures).[1, 2, 3] The crux of the issue lies in the insufficiently diverse training data sets that introduce bias into machine learning algorithms used in all of the applications listed. Popular, publicly available, and easily accessible data sets such as Labeled Faces in the Wild (LFW) are used by software companies (e.g., Google) for benchmarking their facial recognition machine learning algorithms.[4, 2] In 2015, Google's facial recognition app mistakenly identified black people as gorillas which kicked off a conversation about algorithmic bias and its potential implications for humans.[5, 2] Algorithmic bias like that which caused the misidentification of black people can be introduced during training when using data sets such as LFW which Wang et. al. estimated to be nearly 85% white.[6] The resulting model will inherently have more error in identifying underrepresented races, and therefore the algorithm will more accurately predict features of better represented races.

In this work, we evaluate racial algorithmic bias from the standpoint of gender classification using several publicly available data sets. First, we discuss the selection process for the data set and features of the data, the models used herein, and our choice for using gender classification as a benchmark. Then we detail our approach for image pre-processing, data ingestion to the mode, the model architecture, and augmentation of images for model improvement. Finally, we present a comparison of the performance for gender classification with respect to race for each of the methods outlined. Our overarching goal is to reduce algorithmic bias by training a convolutional neural net (CNN) for gender classification using a more diverse training data set, and then test it against an existing non-diverse training data set (e.g., LFW).

1.1 Image Data Selection

A summary of four publicly available facial image databases that we evaluated for this study is given in Table 1. For our purposes, we sought a data set with explicit labels for race and gender. The VGGFace2 data set is the most comprehensive with over 3.3 million faces and more than 8600 identities at different ages.[7] Unfortunately, VGGFace2 is not explicitly labeled for race, and therefore we opted to use the UTKFace data set which was conveniently labeled for both gender and race with a relatively good sample size of 23,000 images.[8] We evaluated LFW and a data set created by a group in China called 'Racial Faces in the Wild' with race labels, but neither data set suited our needs for gender labels.[4, 6]

Table 1. Summary of facial image databases.

Name	# Faces (thousands)	Labels	Ref
VGGFace2	3,300	a,g,i	[7]
LFW	13	i	[4]
RFW	665	r,i	[6]
UTKFace	23	a,g,r	[8]

age (a), gender (g), race (r), identity (i)

The UTKFace data was developed by researchers at the University of Tennessee Knoxville (UTK) who were developing a conditional adversarial autoencoder for age progression/regression and sought a racially and age-balanced data set.[8] As a result, there is a wide range of ages of the people represented in the images. UTKFace contains images that range in age from 1 to 116 with a mean age of 33. Four races and one aggregated category are represented: white, black, asian, indian, and other (hispanic, latino, middle eastern). A comparison of the distribution for age and race with respect to gender is shown in Figure 1. Male and female are represented fairly equally among the races with slightly more male images in the white and indian categories and slightly more female images in the asian and other categories. We are not explicitly

dealing with age for our analysis, but we anecdotally observe that male images between ages 20 to 60 are more evenly distributed whereas there are far more female images between ages 20 and 40. For our analysis, we limit the range of ages to between age 10 to 65.

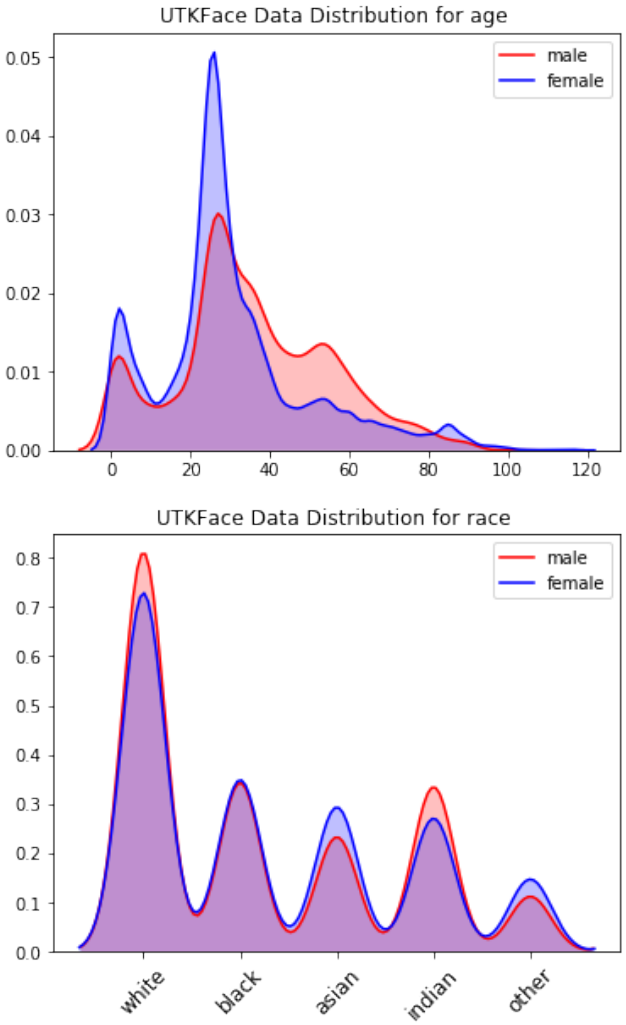

Figure 1. Age and race distribution in the UTKFace data set with respect to the gender.

The UTKFace data set is available in 2 formats: either "in-the-wild" wherein faces are not explicitly cropped and aligned but contain a single face per image or aligned and cropped to a single face. There are readily available software packages that aid in face detection, image alignment, and cropping and therefore we will not discuss those techniques.[9, 10] To accommodate our compressed time frame and to allow us to explore more interesting aspects of model development and fine-tuning, we will use the cropped and aligned UTKFace data set. Eight randomly chosen images from that data set are shown in Figure 2 along with their corresponding age, race, and gender labels at the top of the image. In this subset of images and in general, the labels appear to be correctly assigned for the most part. We did not rigorously evaluate each of the images to verify the labels, but we performed a cursory visual inspection of a few hundred images and found the labels to be correct in more than 95% of the cases.

1.2 Model Selection

We evaluated several publicly available facial recognition models in the hopes of leveraging them with transfer learning to build our gender classifier. Transfer learning is a nuanced practice, and for our purposes we define it as using pre-trained weights and/or the general architecture of an existing model using similar data to perform a new classification task.[11, 12] A comparison of model complexity in terms of the number of parameters for each model is given in Table 2.

Table 2. Summary of transfer learning models.

Model	# Parameters (million)	Ref
VGG	138	[13]
InceptionV3	7	[14]
ResNet50	25.5	[15]

Each of the models listed in Table 2 are available with pre-trained weights for identity classification. For our purposes, we intend to use a neural network model for the classification of gender in a racially diverse data set. Because none of these models are trained for binary classification of gender, we found that they did not do well in classifying gender out-of-the-box. When the models listed in Table 2, one can either use the full model directly or use only the architecture and trainimag using a different data set. Prior to training, the architecture can be manipulated to perform a new task (e.g., gender classification) or to achieve higher accuracy for different data sets.

The models listed in Table 2 are highly complex with millions of parameters each. We decided to evaluate a much simpler model to juxtapose the possibility that a small architecture tuned with a balanced data set can achieve good accuracy for gender classification. Subedi recently developed a CNN with 674,178 parameters for gender, age, and race classification from the UTKFace data set, and the architecture is shown in Figure 3.[16] Subedi's model architecture consists of 5 2D convolution layers each with a batch normalization layer followed by a 2D maximum pooling layer afterwards. The input layer uses a filter size of 32, and each subsequent convolutional layer multiplies the filter size by an integer corresponding to the layer number (e.g., for concolutional layer 2 the filter size is 32×2). The activation function is a rectified linear unit (ReLU) for all convolutional layers. Subedi uses a so-called 'bottleneck' layer (`GlobalMaxPool2D` in Keras) which outputs a 1-dimensional tensor and feeds 3 separate dense layers to produce 3 outputs. In this way, a dense layer is defined for each of the 3 categories each with an appropriate activation function for the type of classification problem (e.g., binary or multiclass). When the model is created, the 3 dense layers are passed in a list as the output. For our purposes, we have adapted the model to output only the gender

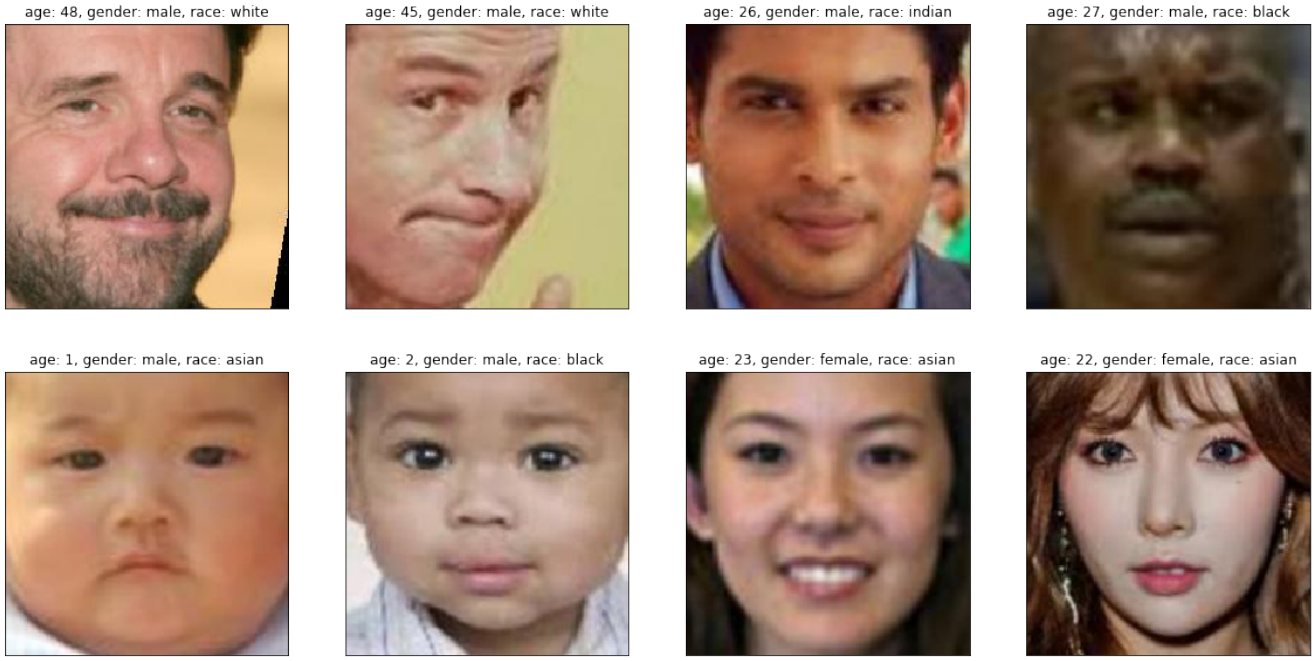

Figure 2. A random sampling of images plotted with their labels shows that the UTKFace data set has very accurate labeling.

classification.

2. Methodology

Keras with a GPU Tensorflow back end was used for implementing and evaluating CNNs, and all models were run using an NVIDIA Tesla K80 GPU. All code written and used from other sources for this report is given in the accompanying iPython Notebook.

2.1 Image Preprocessing

Data ingestion into the model is not straightforward for large data sets, and can create a bottleneck for feeding the neural network that can cause undesirably high computational times. The goal for any data generator is to leverage all available system resources for ingesting data efficiently so that the bottleneck for computation is not the process of feeding data into the model.[17] Keras has a built-in `ImageDataGenerator` class, and we evaluated this for some of the models. Generally, though, we found that results were more comprehensible and predictable using our own custom generator based on the one that was developed by Subedi. We found this uniquely useful when applied with UTKFace data because of the relatively small size of the training data set and the CNN. This allowed us to use a simple data generator that takes in image filenames from a data frame and delivers batches of a user-specified size directly to the model through `fit_generator`.

2.2 Image Augmentation

One drawback to using a custom data generator is that we could not leverage the convenient functions available in the call to Keras' `ImageDataGenerator` for image augementation. As such, we built functionality into our custom generator to add random noise and flip images within the data set as it is ingested into the model. We used the `augmenters` class from within the `imgaug` package to sequentially augment images by adding black pixels to 2% of the image (`CoarseDropout`) and then flipping the image. Model was overfitting when flipping only 30% of the data. When we changed the flip proportion to 50% then the model performed better.

We used over-sampling and down-sampling to balance the data set. The 'other' race had the fewest images (1153) in the training data, and we 'down-sampled' the number of images for the other races to include the same number of images. Over-sampling was done by matching the number of images in the race with the most images (white, 6755). Images of races with fewer than 6755 images were added to the training data set by sampling with replacement. The new over-sampled training data set was then passed via the image data generator to the model. Through that pipeline, the image augmentation wrapped inside the data generator ensured that no two images were identical.

3. Results

Because we began with a broad range of models and data for this work, we began by performing cursory experiments with each model to evaluate 1) computational time, 2) baseline accuracy, and 3) ease of use. Our initial results made it clear that working with the pre-trained models would be computationally prohibitive. We also found that they did not achieve satisfactory results for gender classification.

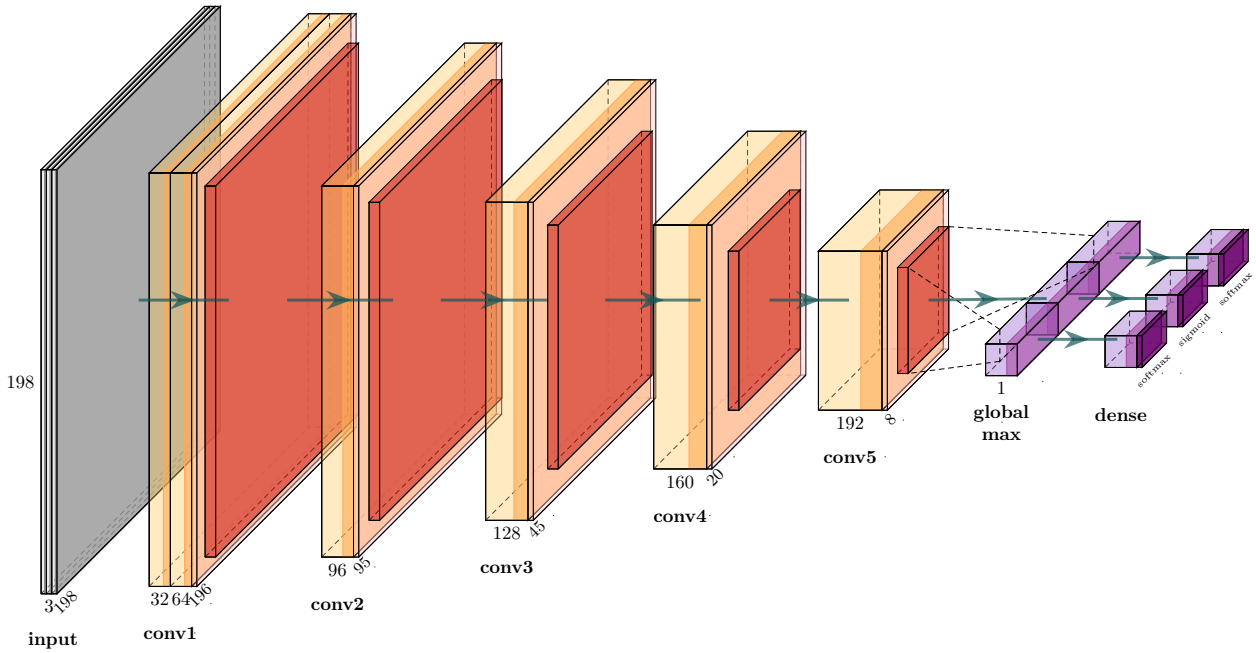

Figure 3. Architecture of neural network developed by Subedi for age, gender, and race classification.

3.1 Gender Classification with Racially Diverse

Performance with respect to age

Used 16 filters Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

3.2 Training with Races Balanced

We took two approaches to balancing the data set: oversampling and down-sampling. Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

3.3 Feature Representation Mapping

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

saliency map discuss how PCA might be used to evaluate features for determining gender, but it's beyond the scope of this project. You might use PCA

3.4 Validation with Other Data Sets

We validated the model using the VGGFace2 and the RFW test data sets. Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

confusion matrix

4. Conclusions

During this project we improved a classifier for gender prediction from racially heterogeneous images. Our accomplishments include:

- list of conclusions

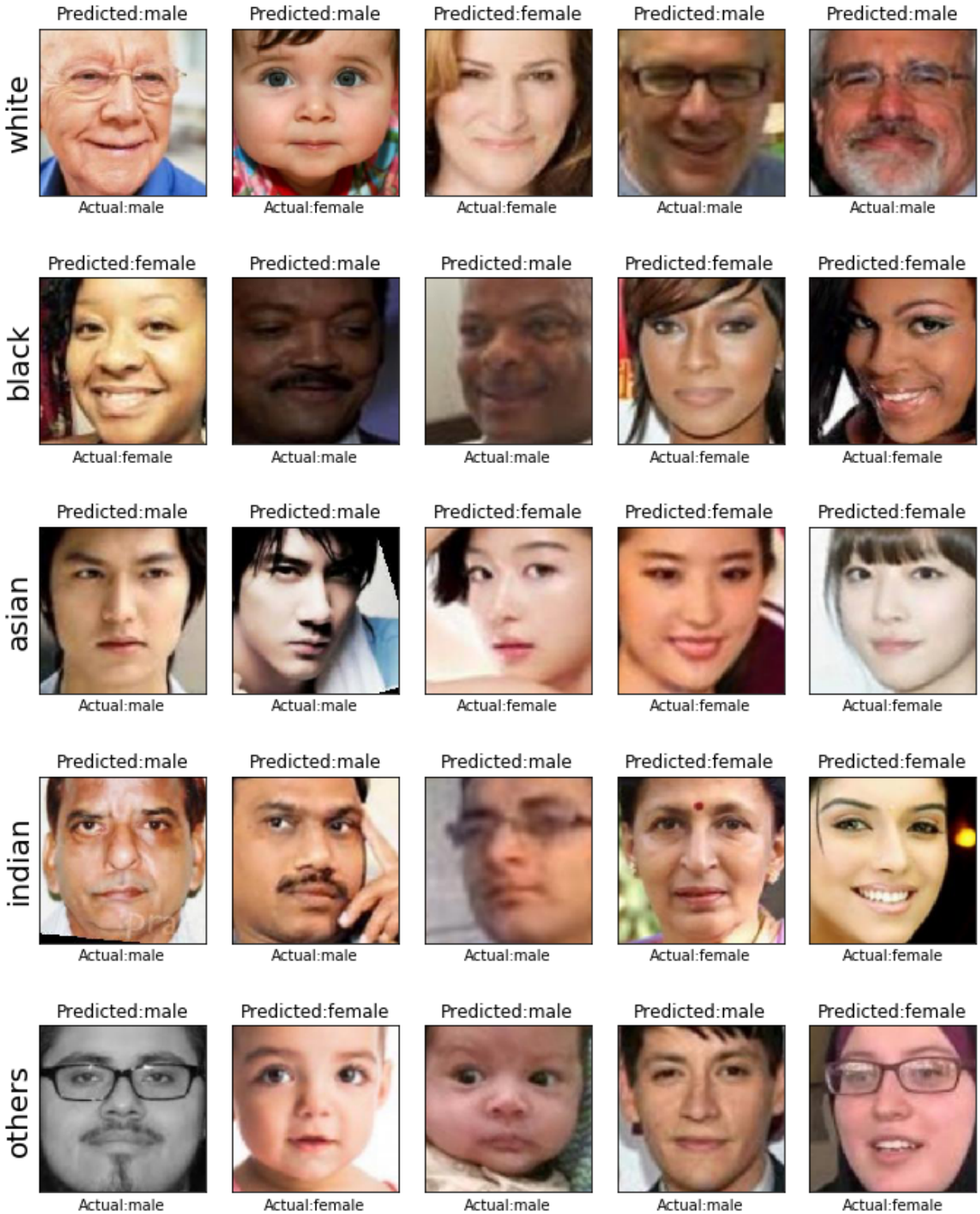

Figure 4. Base case model results with classification results as the top caption and the ground truth as the bottom caption.

- list of conclusions
- list of conclusions

References

[1] C. Couch. *Ghosts in the Machine*, 2017. https://www.pbs.org/wgbh/nova/article/ai-bias/.

[2] S. Lohr. *Facial Recognition is Accurate, if You're a White Guy*, 2018. https://www.nytimes.com/2018/02/09/technology/facial-recognition-race-artificial-intelligence.html.

[3] S. Kim. *Racial Bias in Facial Recognition Software*, 2018. https://blog.algorithmia.com/racial-bias-in-facial-recognition-software/ .

[4] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[5] C. Dougherty. *Google Photos Mistakenly Labels Black People 'Gorillas'*, 2015. https://bits.blogs.nytimes.com/2015/07/01/google-photos-mistakenly-labels-black-people-gorillas.

[6] M. Wang, W. Deng, J. Hu, J. Peng, X. Tao, and Y. Huang. Racial faces in-the-wild: Reducing racial bias by deep unsupervised domain adaptation. *Computer Vision and Pattern Recognition*, 2018. https://arxiv.org/abs/1812.00194 .

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.

[8] Zhang, Zhifei, Song, Yang, , Qi, and Hairong. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[9] D.E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[10] J. Xiang and G. Zhu. Joint face detection and facial expression recognition with mtcnn. In *4th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2017.

[11] J. Brownlee. *A Gentle Introduction to Transfer Learning for Deep Learning*, 2017. https://machinelearningmastery.com/transfer-learning-for-deep-learning/.

[12] D. Sarkar. *A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning*, 2018. https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a.

[13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[14] Szegedy, Vanhoucke, Ioffe, Shlens, and Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[15] He, Zhang, Ren, and Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.

[16] S. Subedi. *Multi output neural network in Keras*, 2018. https://sanjayasubedi.com.np/deeplearning/multioutput-keras/.

[17] A. Amidi and S. Amidi. *A detailed example of how to use data generators with Keras*, 2017. https://stanford.edu/~shervine/blog/keras-how-to-generate-data-on-the-fly.