Estimating Face Demographics using Neural Networks

Abstract

Studying and estimating demographics can be very useful in a variety of situations. Whether it is determining demographics during popular public events or in historical events (to observe fluctuations in demographics due major historical event) or for security purposes in airports using cameras, solving this problem can have numerous applications. We want to utilize neural networks to give us multiple statistics like gender, race and possibly age of people in photos. This problem can be basically broken down into majorly two steps. Firstly, detecting faces in an image and secondly, classifying these faces into multiple demographic categories. Both problems have been individually solved by neural networks with good success. Hence, combining them in a pipeline to study the demographics in a photo is the motive of our project.

1. Introduction

Face recognition has achieved good success. State-of-the-art methods like FaceNet [14] have contributed immensely in these areas. Due to this, it has been possible to use these in a variety of scenarios. In this report, we present one such use case where we extract faces from images containing large number of faces, and then classify them. We target to first classify only for race and gender attributes of a face in this report. We use popular computer vision libraries like OpenCV and dlib [10] for handling the face images.

Literature has a variety of efforts targeting demographic estimation of some form but even state of the art systems fail in recognizing race and gender at times resulting in huge disparity in the classification for certain subgroups. Even though not directly, but incidents such as Google categorizing a black-skinned person as Gorilla [1] shows the inherent complexity of image classification and specifically on human faces. Problem of bias in classification is age old but no single solution has been universal for this. There are various subtle characteristics in this classification that no single model has proved to be completely flawless. We hope to provide a framework for solving this issue using multi-task neural networks. The report is divided in an introduction in section 1. Then, we briefly describe the background and related work in section 2. It is followed by the technical approach in section 3. After that, it is followed by the experimental results which we have obtained till now in section 4. We conclude in section 5 followed by references.

2. Background/Related Work

Literature study on the subject of face recognition and categorization of human faces reveals a lot of complexity and urges the need to deal with this problem. Below we discuss the various aspects of the problem and list few efforts taken in the past on this topic.

2.1. Face recognition/Feature detection

Face Detection and recognition has become a great source of interest after the introduction of Convolutional Neural Networks and has become instrumental in many applications including Face Detection [15], soft biometrics / face attribute classification [3], face recognition [12], emotion analysis [2], gender detection [11], etc. Giants like Google, Amazon and Facebook have developed state of the art facial detection systems and integrated them into their products. Despite huge resources pouring into the field, face recognition and classification for some categories remain low for certain categories, leading to downright blunders like classifiying black women as Gorilla [1]. Google tried to bypass this issue by removing certain labels like Monkeys, Gorillas, etc. completely from their classification model. Particularly individuals belonging to categories like female, Black people and people between the age of 18 and 30 years exhibit low accuracy for face recognition systems used in US-based law enforcement, as reported by Klare et al. [4].

Farinella and Dugelay [8] alleged that ethnicity has no effect on gender classification, adopting a binary ethnic categorization scheme for the experimentation: Caucasian and non-Caucasian. Dwork et al. [6] demonstrated the importance of understanding sensitive characteristics such as gender and race, in order to build demographically inclusive models. Fairness in these models can be achieved by having equal representation of each subgroups during

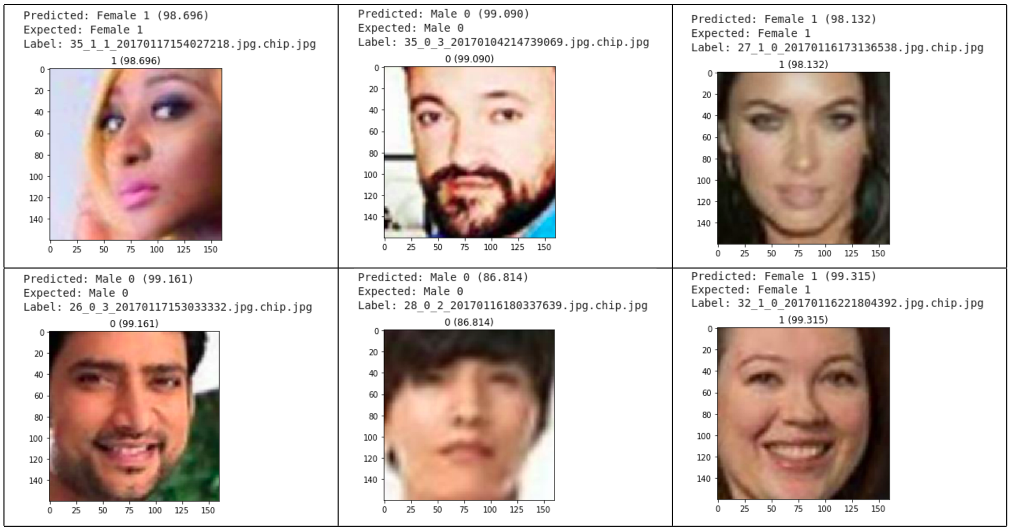

Figure 1. Sample gender classification results. The Predicted field shows the predicted label and the probability assigned to this label by the network in brackets. The Expected field shows the expected label for this image and the Label field shows the manual labels present in the filename of the image.

training the model. Recent work by Buolamwini et al. [5] showed that among the 3 commercial gender classification systems, darker-skinned females are the most mis-classified group. This is a clear cut case of bias in classification systems. To reduce this bias, Ryu et al. [13] has introduced the usage of multi-task learning networks for better classification models. Similar model of MTCNN was employed by Das et al. [7]

Hence, we see specific challenges in neural networks being able to distinguish facial features of humans from face images in a straightforward way. Human faces have very minor differences and thus a straightforward approach of using CNNs for image classification of different objects might not be very useful when dealing with finer intricacies of human facial features. These complexities and the applicability of facial demographic study to a wide variety of scenarios motivated us to take up the topic and investigate these systems and possibly make improvements, if any.

2.2. Datasets

We wanted to select an image dataset which contains labels with gender and race labels. Few database have very high bias towards few categories of demograhics. For example, Labeled Faces in the Wild (LFW) contains 77.5% male and 83.5% Caucasians [9]. To mitigate these limitations, Intelligence Advanced Research Projects Activity (IARPA) introduced an initiative to release the IJB-A dataset as the most diverse set. However, this dataset is not annotated with age, which might be helpful in ascertaining certain features down the lane.

Hence, we chose to work with UTKFace dataset which has most of the required features, including diverse set of population, annotations for gender, age, ethnicity. The dataset is a large-scale with long age span (range from 0 to 116 years old) and consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. It also provides a smaller set of cropped and aligned images which was helpful for us to begin without a lot of pre-processing. Since the future extensions might need features that are tied to age, we would want a database which annotates age as well. Hence, we chose to go with UTKFace, which is a well-referenced standard benchmark for systems performing similar tasks.

3. Approach

For performing the demographics estimation, we divide the entire effort into two stages, face extraction and face classification. The following sub-sections describe in detail about both the stages and the progress we have made with each of these. We also enlist the steps which are taken in each stage and which develop a flow to make the entire pipeline for final stitching of the stages.

3.1. Face Extraction

Face extraction is the process where we detect all the faces in the group images and then extract these out to give us face images for the classification stage.

- We use OpenCV to read images and matplotlib for all the plots.

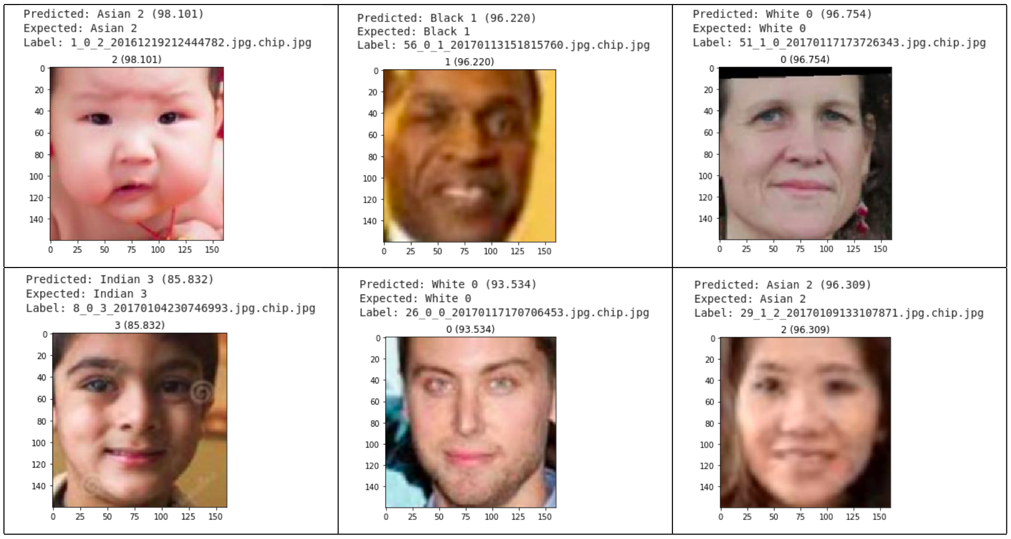

Figure 2. Sample race classification results. The Predicted field shows the predicted label and the probability assigned to this label by the network in brackets. The Expected field shows the expected label for this image and the Label field shows the manual labels present in the filename of the image.

- We use face landmarks model designed for dlib library [10] (which uses combination of HOG and SVM for detection) for pre-processing and extracting faces from group images.
- OpenFace, an open-source implementation of FaceNet [14] provides this dlib alignment feature for pre-processing and we use it here. The implementation results into bounding boxes over the faces.
- The faces inside the bounding boxes are extracted as face images. We process these to be fed into our classification system in next step.
- We perform alignment and resizing of the face images. Faces are resized to the same size (160px x 160px in our case) and transformed to make landmarks (such as the eyes and nose) appear at the same location on every image.
- We manually look at these faces and give them gender and race labels by saving the extracted face images with these labels in their filenames. These manual annotations are used for measuring the performance of our classification framework on these extracted testing images. For example, each extracted face image filename contains the group image it is part of, the index of extraction and the gender and race labels.

3.2. Face Classification

3.2.1 Initial Approach - Transfer Learning

We perform face classification on various demographic attributes in this stage. The process is a general neural network model training and then using it to find the demographic labels for the unseen images.

- Training images for classification are obtained from UTKFace dataset.
- Labeling was done for gender and race together totaling around 10 classes, 5 races for each of the 2 genders.
- Cropped and aligned images are converted to numpy arrays and corresponding labels are extracted and assigned.
- Since the number of examples for each class of images is very skewed, initial results were not good. So, we used data augmentation (like flipping horizontally, grayscale of images) to increase the number of images for classes with low number of examples and dropped few examples from high density classes.
- We used Keras Library over tensorflow for classifying images using Transfer Learning. Here we used ResNet50 followed by fully connected layers for gender and ethnicity detection. We used pre-trained weights for ImageNet in Keras and used dense connected layer for classification.
- We tried to implement the fully connected layer in two ways. Firstly, trying to train using combined 10 labels and then using separate networks for both race and gender. Refer to figure 3 for the initial architecture.

Limitations This approach has few fundamentals problems noted below:

- Since the resnet50 is deep, the model overfits for our dataset.

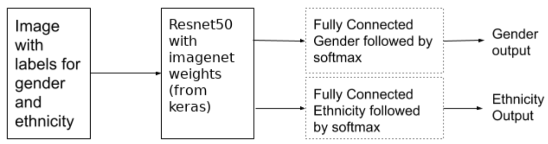

Figure 3. Architecture of the initial approach

- While using data augmentation, we added grayscale images as well to see if the model can learn contrasting features but this turned out to be performing worse on the validation data.
- Using combined labels for race and gender turned out to give less accuracy even after lot of tuning. Hence, we decide to separate the category labels for gender and race in our final approach. Literature survey helped us discover MTCNN which we utilize to improve our efforts which is described in the following sections.

Figure 4. Recognizing faces using detection

Figure 5. Images extracted using detection

3.2.2 Final Approach - MTCNN

After trying our initial approach, we decided to adopt another technique. We found literature using MTCNN converting faces to embeddings and then classifying them. The Figure 7 shows the final architecture used. Also, we describe few important terms below.

Multi-Task Convolutional Neural Networks (MTCNN): Most of the available face detection and face alignment methods ignore the inherent correlation between these two tasks. Zhang et al. [17] proposed a new framework to improve the accuracy for multi-view face detection called Multi-Task Deep Convolutional Neural Network capable of simultaneously learning to detect face/non-face, face pose and facial landmarks through multi-task view in the last layer. The architecture and claims of improvement have been explored in the original paper [18].

Face Embedding: The Face Embedding model analyzes images and returns numerical vectors in Euclidean space that represent each face as n-dimensional vector. The vectors of visually similar faces will be close to each other in the space. The Face Embedding model can be used for organizing, filtering, and ranking images according to visual similarity. Referring the FaceNet paper [14] can help develop better understanding of the generation of these vectors. For our case, the vector representation is computed by using pre-trained FaceNet model weights. The MTCNN model using the FaceNet weights generates face embeddings in 128 dimensions.

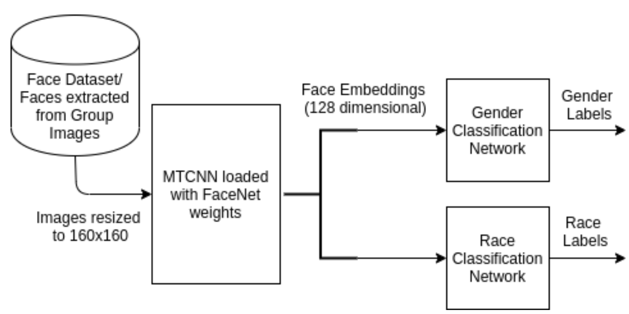

Figure 7. Architecture of the initial approach

4. Experiments and Results

4.1. Face Recognition/Detection

1. Face Image Extraction using OpenFace pre-trained landmarks model

    - Since we are using pre-trained face landmarks

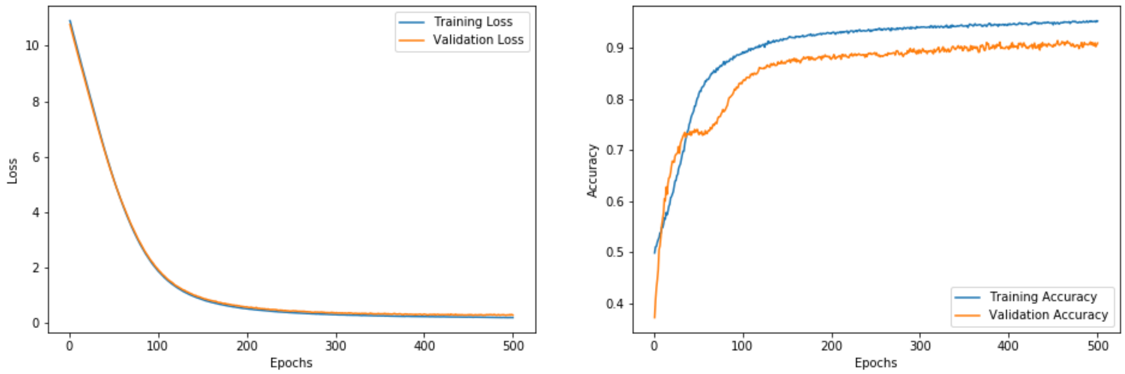

Figure 6. Classification accuracy for gender classification

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 100)	12900
batch_normalization_1 (Batch	(None, 100)	400
activation_1 (Activation)	(None, 100)	0
dropout_1 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 100)	10100
batch_normalization_2 (Batch	(None, 100)	400
activation_2 (Activation)	(None, 100)	0
dropout_2 (Dropout)	(None, 100)	0
dense_3 (Dense)	(None, 100)	10100
batch_normalization_3 (Batch	(None, 100)	400
activation_3 (Activation)	(None, 100)	0
dropout_3 (Dropout)	(None, 100)	0
dense_4 (Dense)	(None, 100)	10100
batch_normalization_4 (Batch	(None, 100)	400
activation_4 (Activation)	(None, 100)	0
dropout_4 (Dropout)	(None, 100)	0
dense_5 (Dense)	(None, 100)	10100
batch_normalization_5 (Batch	(None, 100)	400
activation_5 (Activation)	(None, 100)	0
dropout_5 (Dropout)	(None, 100)	0
dense_6 (Dense)	(None, 2)	202
batch_normalization_6 (Batch	(None, 2)	8
activation_6 (Activation)	(None, 2)	0

Total params: 55,510
Trainable params: 54,506
Non-trainable params: 1,004

Figure 8. Architecture of gender classification model

Layer (type)	Output Shape	Param #
dense_27 (Dense)	(None, 200)	25800
batch_normalization_27 (Batc	(None, 200)	800
activation_27 (Activation)	(None, 200)	0
dropout_22 (Dropout)	(None, 200)	0
dense_28 (Dense)	(None, 100)	20100
batch_normalization_28 (Batc	(None, 100)	400
activation_28 (Activation)	(None, 100)	0
dropout_23 (Dropout)	(None, 100)	0
dense_29 (Dense)	(None, 100)	10100
batch_normalization_29 (Batc	(None, 100)	400
activation_29 (Activation)	(None, 100)	0
dropout_24 (Dropout)	(None, 100)	0
dense_30 (Dense)	(None, 100)	10100
batch_normalization_30 (Batc	(None, 100)	400
activation_30 (Activation)	(None, 100)	0
dropout_25 (Dropout)	(None, 100)	0
dense_31 (Dense)	(None, 5)	505
batch_normalization_31 (Batc	(None, 5)	20
activation_31 (Activation)	(None, 5)	0

Total params: 68,625
Trainable params: 67,615
Non-trainable params: 1,010

Figure 9. Architecture of race classification model

and use dlib alignment, we are able to detect faces in group images very well.

- The dlib alignment ensures that face features are well aligned and hence ready for next stage of face classification.
- Since we have directly used an existing model for this, we are only reporting qualitative results for this part of the project. You can find examples in Figure 4 and Figure 5. Any inaccuracies are inherent to the FaceNet model being capable of generating bound boxes directly.

4.2. Face Classification

4.2.1 Dataset

As mentioned before we chose to go with UTKFace, which is a standard benchmark for face classification.

- **UTKFace** dataset is a large-scale face dataset (over 20,000 face images) with a population coverage of long age span ranged from 0 to 116 years.
- Specifically, annotation for age includes following classes: baby: 0-3 years, child: 4-12 years, teenagers:

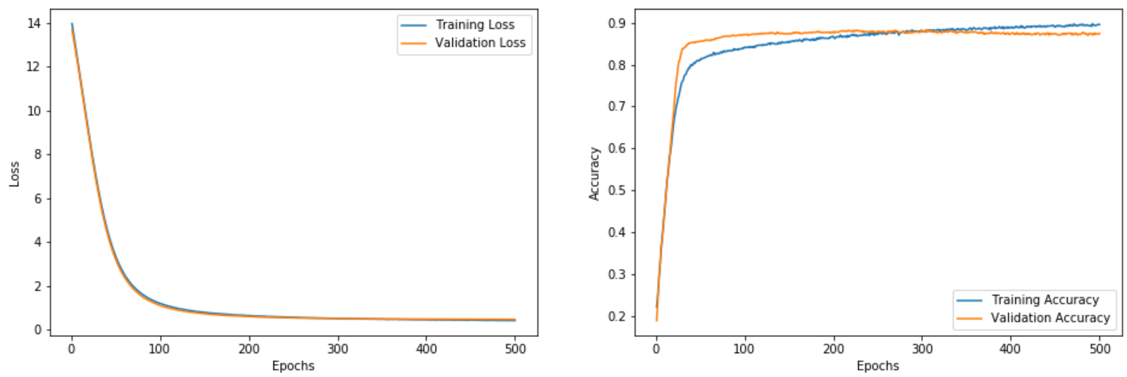

Figure 10. Classification accuracy for race classification

13-19 years, young: 20-30 years, adult: 31-45 years, middle aged: 46-60 years and senior: 61 and above years.

- The dataset additionally contains the labeling for gender (male and female), as well as five races (White, Black, Asian, Indian and other race).
- UTKFace includes large variations of pose, facial expression, illumination, occlusion, and resolution.

4.2.2 Experimentation Protocol

We classify gender and race, respectively, analyzing the trait instances of the remaining facial attributes.

- The dataset consists of three parts. Part I, II and III consist of 10437, 10719 and 3252 images, respectively.
- The weights are initialized with He normal (he-et-al initialization) and have L2 regularization with regularization strength of 0.01 on all dense layer weights.
- The learning rate we settled upon is 1e-6 for both classifiers with a learning rate decay of 1e-6 for race classifier.
- We used ReLU activation layer after each dense layers, except the last layer where we use softmax layer.
- We have batch normalization before each non-linearity to keep the inputs to non-linearity normalized.
- To avoid classifier networks to overfit, we had to add dropout before next dense layer. We have varying dropout rates of 0.2 and 0.3 in the hidden layers.
- Tried different types of optimizers including Stochastic Gradient Descent, RMSprop and Adam. We used Adam for gender classifier, however we settled for RMSprop for race classifier as it allowed the network to learn in a way that is not as slow as SGD (to keep the number of epochs less) and not as fast as Adam (to allow network to learn finer details).
- Implemented mini-batching with varying batch sizes and settled for the batch-size 64 for gender classifier and 32 for race classifier.
- We used is categorical crossentropy offered by keras for the loss function.
- The final dense layer has the number of neurons equal to the number of classes (2 for gender and 5 for race classifiers).
- The final architectures of gender and race classification networks are shown in figures Figure 8 and Figure 9 respectively.
- For the gender classifier, we used 21333 images for training with a cross-validation split of 20% and 2372 images for testing. For the race classifier, we used same division of training, validation and testing images.

4.2.3 Results and analysis

Gender Classification

After the training phase, the baseline model's performance on test data comprising of 2372 images was measured and the following metrics are reported:

1. Validation loss and accuracy in Table 1
2. Testing confusion matrix in Table 2
3. Testing accuracy for each class in Table 3
4. Graph for loss and accuracy (comparing training and validation) in Figure 6
5. Sample results of classification predicted labels are shown in Figure 1

Metric	Results
Loss	0.2707
Accuracy	91%

Table 1. Validation Performance of Gender Classification

As you can see from the Table 2, the confusion matrix for gender classification is showing the results for each classification on test data, and the class-wise accuracy is shown in Table 3.

	Male	Female
Male	1160	80
Female	57	1075

Table 2. Testing Confusion matrix for Gender Classification

Class	Accuracy%
Male	93.55
Female	94.96

Table 3. Testing class-wise accuracy for Gender Classification

Race Classification

After the training phase, the baseline model's performance on test data comprising of 2373 images was measured and the following metrics are reported:

1. Validation loss and accuracy in Table 4
2. Testing confusion matrix in Table 5
3. Testing accuracy in Table 6
4. Graph for loss and accuracy (comparing training and validation) in Figure 10
5. Sample results of classification predicted labels are shown in Figure 2

Metric	Results
Loss	0.45
Accuracy	87.43%

Table 4. Validation Performance of Race Classification

As you can see from the Table number 2, the confusion matrix for race classification is showing the results for each classification on testing data and the class-wise accuracy is shown in Table 6.

We were able to achieve **test accuracy** of **94**% accuracy for gender classification and **87**% accuracy for race classification. In Figure 11, we show few examples results of faces extracted in Figure 5 classified by our network.

	White	Black	Asian	India	Others
White	928	18	5	29	28
Black	8	422	3	11	9
Asian	16	1	319	5	3
India	24	8	3	351	12
Others	68	11	11	26	54

Table 5. Testing confusion matrix for Race Classification

Class	Accuracy%
White	92.06
Black	93.15
Asian	92.73
India	88.19
Others	31.76

Table 6. Testing class-wise accuracy for Race Classification

One interesting observation we made here is the bias in the network for different classes. We see that the race classification showed very less accuracy for the 'Others' race. We accredit this to two reasons, firstly due to less data for this class and secondly, due to very less similarity in the faces in this class as it contains images of people from any races in the world other than the first four classes. We also see how balanced data size for every class can help keeping the network generalized overall and give good performance in classifying in those classes.

5. Conclusion and Future Work

Our goal is to estimate face demographics by identifying different races and gender of faces in pictures with a lot of faces. Our future plan is to improve the model to get better validation and test accuracies and apply it in various contexts to see if we could make meaningful deductions. We accredit the reason for the accuracy being less in our case to the unbalanced nature of all classes in UTKFace dataset. As seen in one of the observations, a class has less classification accuracy because of less training images in ratio to other classes. To improve this, we can perform data augmentation of faces for that class. However, data augmentation techniques to generate human face images is not very straightforward. Generic augmentation techniques like flipping or shifting can make a face lose the features which identify it under certain labels like it's race. Thus, facial data augmentation is still an active area of research and there have been studies to compare existing augmentation efforts [16]. However, not many standard techniques are available to be used directly.

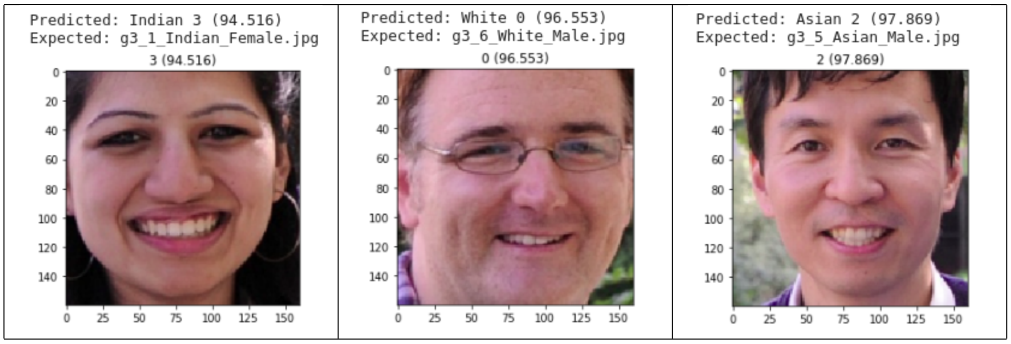

Figure 11. Classification of detected faces from Figure 5. The Predicted field contains label and the probability assigned to it in bracket. The Expected field shows the filename containing manually assigned labels to the extracted face.

One major learning of the whole project was that often training a neural network isn't always straightforward. Finding clues to it's behaviour might not be explicit. For example, during our initial trials, tiny changes to learning rate diverged the network completely. Increasing the depth of the network helped stabilize it better. Another learning is that it's often very easy to overlook the inherent biases introduced in the model involuntarily because of data itself. For example, if the number of examples from one subgroup is significantly more than the other (just like in our case), it might lead to the larger size class predicted often. Cross-validation can help notice this behaviour. With enforcement agencies, governments and large corporate entities using face recognition frameworks on daily basis, these models must be kept free of bias to prevent unintended consequences.

One more learning is that we need to experiment with lots of hypothesis when training neural networks and refine any assumptions as we encounter results different from expected. In our case, we tried data augmentation by adding grayscale images. We hoped that the contrast features extracted from this would be helpful for the model to classify faces better. But this was in-turn throwing away some features (like skin color) and performance worsened. Adding grayscale images isn't always bad but it wasn't suitable for our usecase. Conclusion is that experimentation is key in neural networks.

References

[1] https://www.wnycstudios.org/podcasts/notetoself/episodes/deep-problem-deep-learning.

[2] G. S. S. Z. M. Afshin Dehghan, Enrique G. Ortiz. Dager: Deep age, gender and emotion recognition using convolutional neural network. *arXiv preprint*, 11:441–467, 2017.

[3] P. E. Antitza Dantcheva and A. Ross. What else does your biometric data reveal? a survey on soft biometrics. *IEEE*, 11:441–467, 2016.

[4] J. C. K. R. W. V. B. A. K. J. Brendan F. Klare, Mark J. Burge. Face recognition performance: Role of demographic information. *IEEE*, 7:1789–1801, 2012.

[5] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. 81:77–91, 23–24 Feb 2018.

[6] T. P. O. R. R. Z. Cynthia Dwork, Moritz Hardt. Fairness through awareness. *arxiv*, 2011.

[7] A. Das and A. Dantcheva. Mitigating bias in gender, age, and ethnicity classification: a multi-task convolution neural network approach. 10 2018.

[8] G. Farinella and J.-L. Dugelay. Demographic classification: Do gender and ethnicity affect each other? *IEEE*, 2012.

[9] H. Han and A. Jain. Age, gender and race estimation from unconstrained face images. 07 2014.

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

[11] Y. W. Michal Kosinski. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *Journal of Personality and Social Psychology*, 114:246–257, 2018.

[12] C. D. C. Rajeev Ranjan, Swami Sankaranarayanan and R. Chellappa. An all-in-one convolutional neural network for face analysis. *IEEE*, 137:1–24, 2017.

[13] H. J. Ryu, M. Mitchell, and H. Adam. Improving smiling detection with race and gender diversity. *CoRR*, abs/1712.00193, 2017.

[14] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015.

[15] Z. Z. Stefanos Zafeiriou and C. Zhang. A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*, 138:1–24, 2015.

[16] X. Wang, K. Wang, and S. Lian. A survey on face data augmentation. *CoRR*, abs/1904.11685, 2019.

[17] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. *IEEE*, 10:1755–1758, 2014.

[18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.

SUPPLEMENTARY MATERIAL

Search for Optimal Hyper-parameters


```
[{"batch_size": 32, "epochs": 20, "lr": 0.001, "optimizer": "SGD"},
{"batch_size": 32, "epochs": 20, "lr": 0.001, "optimizer": "Adam"},
{"batch_size": 32, "epochs": 20, "lr": 0.001, "optimizer": "RMSprop"},
{"batch_size": 32, "epochs": 20, "lr": 0.0001, "optimizer": "SGD"},
{"batch_size": 32, "epochs": 20, "lr": 0.0001, "optimizer": "Adam"},
{"batch_size": 32, "epochs": 20, "lr": 0.0001, "optimizer": "RMSprop"},
{"batch_size": 32, "epochs": 20, "lr": 1e-05, "optimizer": "SGD"},
{"batch_size": 32, "epochs": 20, "lr": 1e-05, "optimizer": "Adam"},
{"batch_size": 32, "epochs": 20, "lr": 1e-05, "optimizer": "RMSprop"},
{"batch_size": 32, "epochs": 20, "lr": 1e-06, "optimizer": "SGD"},
{"batch_size": 32, "epochs": 20, "lr": 1e-06, "optimizer": "Adam"},
{"batch_size": 32, "epochs": 20, "lr": 1e-06, "optimizer": "RMSprop"},
{"batch_size": 64, "epochs": 20, "lr": 0.001, "optimizer": "SGD"},
{"batch_size": 64, "epochs": 20, "lr": 0.001, "optimizer": "Adam"},
{"batch_size": 64, "epochs": 20, "lr": 0.001, "optimizer": "RMSprop"},
{"batch_size": 64, "epochs": 20, "lr": 0.0001, "optimizer": "SGD"},
{"batch_size": 64, "epochs": 20, "lr": 0.0001, "optimizer": "Adam"},
{"batch_size": 64, "epochs": 20, "lr": 0.0001, "optimizer": "RMSprop"},
{"batch_size": 64, "epochs": 20, "lr": 1e-05, "optimizer": "SGD"},
{"batch_size": 64, "epochs": 20, "lr": 1e-05, "optimizer": "Adam"},
{"batch_size": 64, "epochs": 20, "lr": 1e-05, "optimizer": "RMSprop"},
{"batch_size": 64, "epochs": 20, "lr": 1e-06, "optimizer": "SGD"},
{"batch_size": 64, "epochs": 20, "lr": 1e-06, "optimizer": "Adam"},
{"batch_size": 64, "epochs": 20, "lr": 1e-06, "optimizer": "RMSprop"},
{"batch_size": 128, "epochs": 20, "lr": 0.001, "optimizer": "SGD"},
{"batch_size": 128, "epochs": 20, "lr": 0.001, "optimizer": "Adam"},
{"batch_size": 128, "epochs": 20, "lr": 0.001, "optimizer": "RMSprop"},
{"batch_size": 128, "epochs": 20, "lr": 0.0001, "optimizer": "SGD"},
{"batch_size": 128, "epochs": 20, "lr": 0.0001, "optimizer": "Adam"},
{"batch_size": 128, "epochs": 20, "lr": 0.0001, "optimizer": "RMSprop"},
{"batch_size": 128, "epochs": 20, "lr": 1e-05, "optimizer": "SGD"},
{"batch_size": 128, "epochs": 20, "lr": 1e-05, "optimizer": "Adam"},
{"batch_size": 128, "epochs": 20, "lr": 1e-05, "optimizer": "RMSprop"},
{"batch_size": 128, "epochs": 20, "lr": 1e-06, "optimizer": "SGD"},
{"batch_size": 128, "epochs": 20, "lr": 1e-06, "optimizer": "Adam"},
{"batch_size": 128, "epochs": 20, "lr": 1e-06, "optimizer": "RMSprop"}]
```


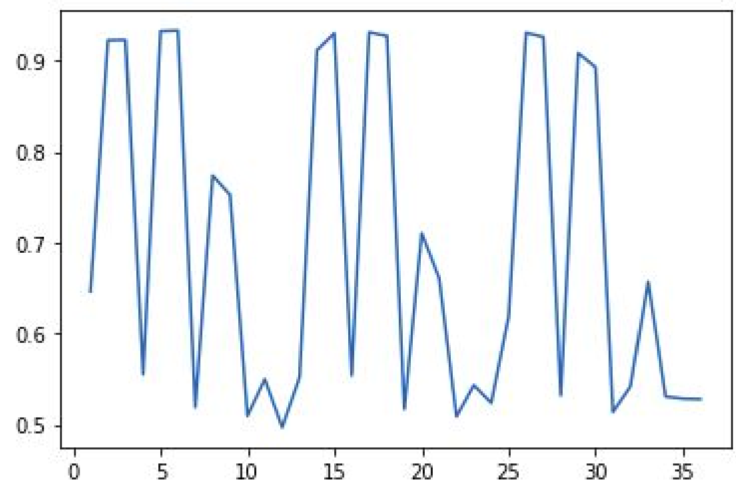

Figure. Validation accuracy vs Index in Param Dictionary above

SUPPLEMENTARY MATERIAL

Face Recognition and Extraction Results

Few Examples of Extracted Faces and Assigned Labels