Dimensionality Reduction for Face Generation and Feature Analysis

Colin Siles

December 9th, 2020

Images of human faces appear to be quite complex, given the wide array of possible facial features and expressions. However, due to the shared similarities of human faces, they are an excellent candidate for applying dimensionality reduction. In order to generate random human faces, and understand both the number and types of features present in a face, a series of dimensionality reduction models were created. The models yielded the conclusion that about 128 features are needed to represent a 100x100 image of a human face, and that many of the most important features encoded lighting and gender-related facial structure. The resulting models also showed potential to be used in a variety of applications, including image denoising, face generation, and face modification. The autoencoder was moderately accurate at reconstructing human faces, failing to model finer details of the face, but still producing images that were clearly recognizable as human faces.

Overview

For my semester project, I was interested in exploring unsupervised learning through dimensionality reduction. One type of data that was particularly interesting to me were pictures of faces, because although they are complex, all faces have a similar structure that should be possible to represent with a lower dimensional feature vector. Through dimensionality reduction, I expected to be able to answer questions like, what types and how many features define a face? What are the relative importance of these features? Furthermore, by employing an autoencoder, it would be possible to generate faces from the feature vector, so I would be able to answer questions like how well do these features really represent a human face?

I planned to use UTKFace dataset [1] to complete this project. The dataset consists of 20,000 faces, and claims to have variety in age, gender, ethnicity, facial expression, illumination, and more. Compared to other face datasets, UTKFace offers many benefits, including well-cropped faces, a moderate number of images, and good variety. These factors will make it easier to quickly build models and analyze data, and will also help the resulting models generalize to a variety of different faces. Furthermore, the dataset provides labels for age, gender, and ethnicity, which will allow for analysis to determine if such features are encoded or strongly correlated with components of the feature vector. It also enables age progression/regression applications.

Data Acquisition

I used the UTKFace dataset [1], specifically the data provided by Kaggle [2], which consists of more well-cropped faces than the original dataset.

There are no details about where specifically the images in the dataset come from. However, some images have stock photo watermarks overlaid, suggesting that many of the images may have simply been pulled from various online sources. The Kaggle dataset does not provide a license, but the original dataset mandates non-commercial research purposes only. My use of the dataset for this project should therefore be permissible.

The dataset consists of a single directory of about 20,000 jpeg images of human faces, each with a resolution of 200x200 (although some pictures appear more blurry than others). The age, gender, and race labels are encoded in the name of each image, and the UTKFace website provides details on how to decode these labels. I decided to use all images in the dataset, expecting the variety provided by the full dataset to help the model to generalize better, while also acknowledging that unclean or imbalanced data could interfere with the results.

Preprocessing

To determine the accuracy of the labels, 100 random images were selected from the dataset, and the labels were verified. Of the 100 images, all the labels seemed reasonable, except for the one shown in **Figure 1**, which seems to be incorrect. This low incidence of incorrect labels suggests that most of the data is correctly labeled. For that reason, it does not seem worth the effort to sort through the more than 20,000 images to remove any incorrect labels.

Figure 1

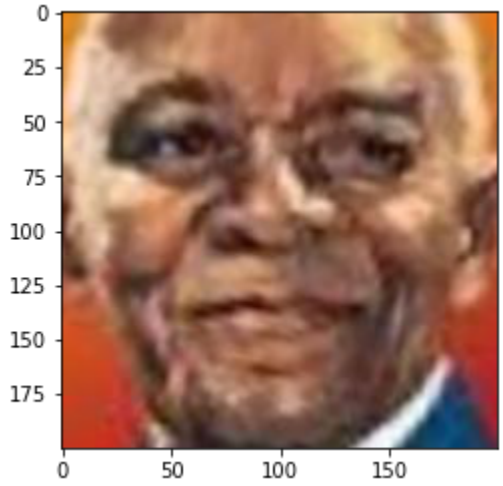

Figure 1 shows the single sample image out of 100 that appeared to be labeled incorrectly. The label for this image was 26 years old, male, and white.

The 100 random images selected also revealed other qualitative features of the data that may pose problems. Primarily, although each image is 200x200, some images are much blurrier than others. **Figure 2** shows a comparison between one of the most clear, and one the most blurry images in the sample of 100 images. The blurriness of most other images sampled falls between these two. Because facial features are still generally identifiable in the blurriest images, it does not seem worth the effort to eliminate any images due to blurriness.

Figure 2

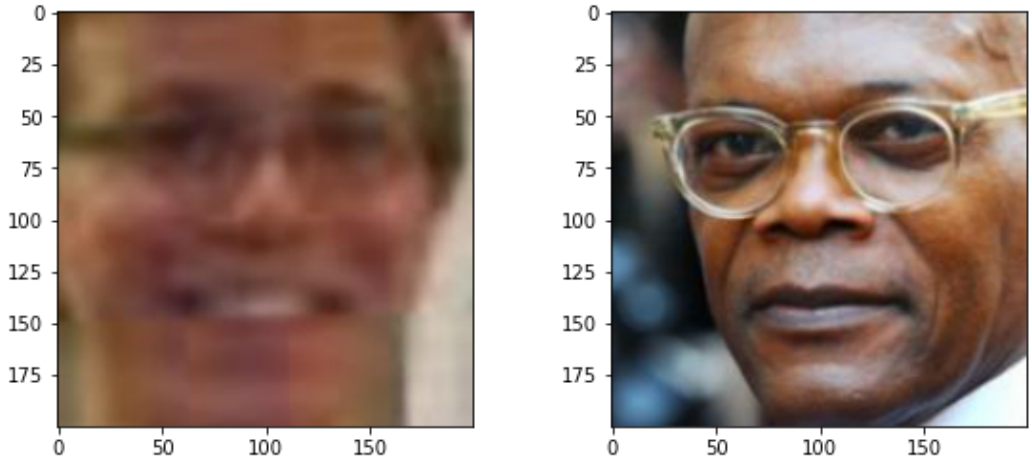

Figure 2 shows the most blurry (left) and most clear (right) samples of the 100 images

---

Other potential problems revealed from the sample of 100 images include watermarks or other visual artifacts superimposed on top of the image (4 of the 100 images), and black and white images (1 of the 100 images). Examples of these are shown in **Figure 3**. Because these problems are difficult to efficiently remove from the dataset, these images were still used.

Figure 3

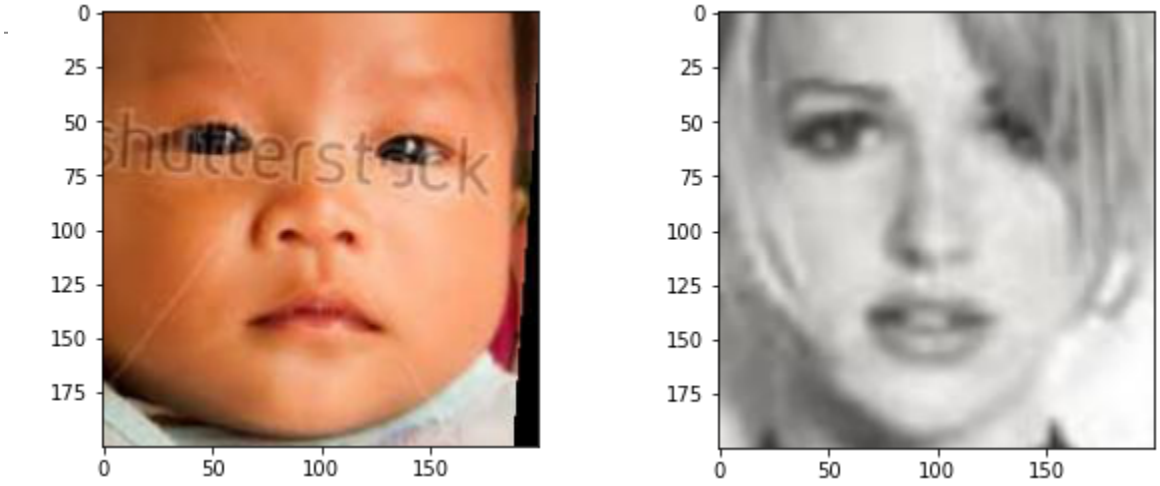

Figure 3 shows a sample of some other potential problems with the dataset, including a watermark superimposed over the face (left) and a black and white face (right)

In addition to a qualitative analysis of the images, a quantitative analysis of the labels was performed. Under the assumption that most of the labels are correct, this analysis served to reveal more information about the composition of the images, beyond data cleansing purposes. An initial analysis revealed that four of the images in the dataset have invalid file names (missing one or more of their labels). For simplicity, these four images were not used.

A broad analysis of the labels revealed a somewhat imbalanced dataset. Although gender is split roughly equally, there are more than twice as many white faces than any other race in the dataset. Although representation of multiple races in the dataset is better than only one, the model may still be biased towards white faces as a result. Furthermore, there are distinct peaks in certain ages, specifically at twenty-six (about 2,200 images) and one (about 1,600 images) year old. Similar to race, although the wide range of ages in the dataset may be beneficial, the imbalance in ages may produce bias in the final model. See **Figures 4-6** for charts that demonstrate these distributions. Because it can often be difficult to determine the optimal data balance before training a model, all of the data was used. Bias towards any specific age, gender, or race was studied for the resulting model.

Figure 4

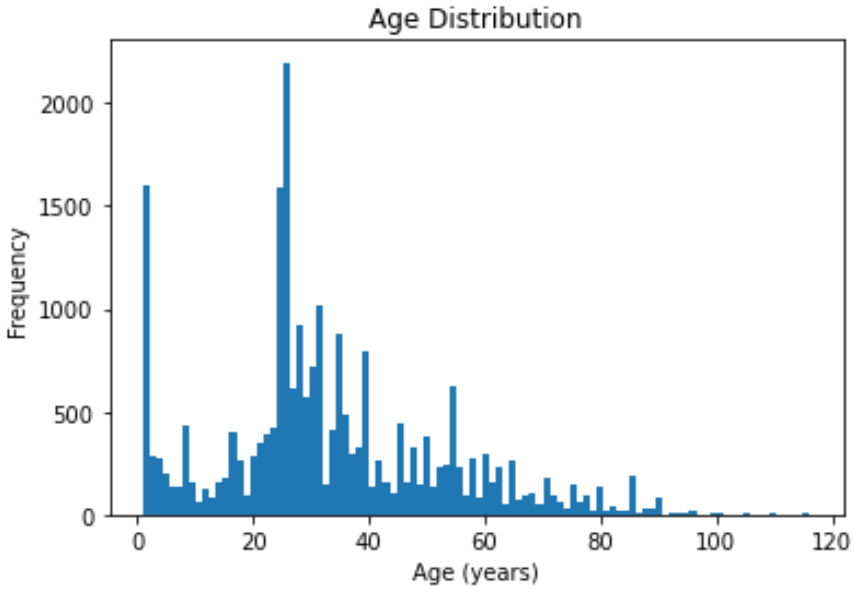

Figure 4 shows the distribution of age labels in the dataset. Note that there are distinct peaks at 1 and 26 years old, and that the distribution is generally not very uniform, although it does represent ages from 1 to 116.

Figure 5

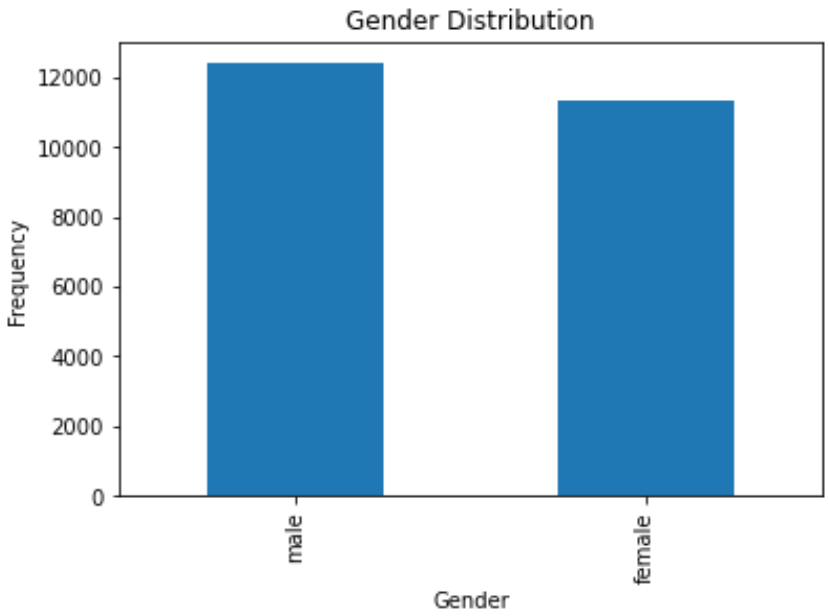

Figure 5 shows the distribution of gender labels in the dataset. Note that despite males are slightly more prevalent, the proportion of male/female is relatively equal

---

Figure 6

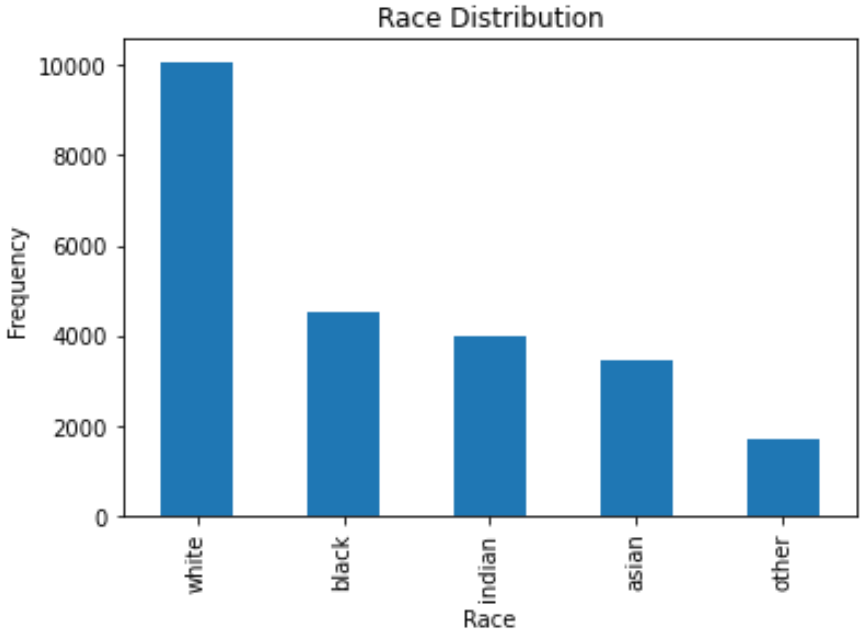

Figure 6 shows the distribution of race labels in the dataset. Note that although there is representation of multiple races, there are significantly more whites than any other single race

A finer analysis of the labels also reveals some minor associations between different labels. In particular, different races tend to represent different age ranges. Although whites have a relatively uniform distribution of ages, blacks are heavily clustered in the 20-40 age range, and asian and indians are skewed towards the younger age ranges. See **Figure 7** for charts that demonstrate this distribution. It is likely that the model will learn these associations between age and race, which may introduce bias into the model (e.g. faces with more "asian" features may be classified as younger). However, similar to the imbalance in the dataset as a whole, it is difficult to determine the appropriate balance to eliminate this correlation, and as such, all the data was still used. Adjustment could be made as necessary after an initial analysis of bias in the model.

Figure 7

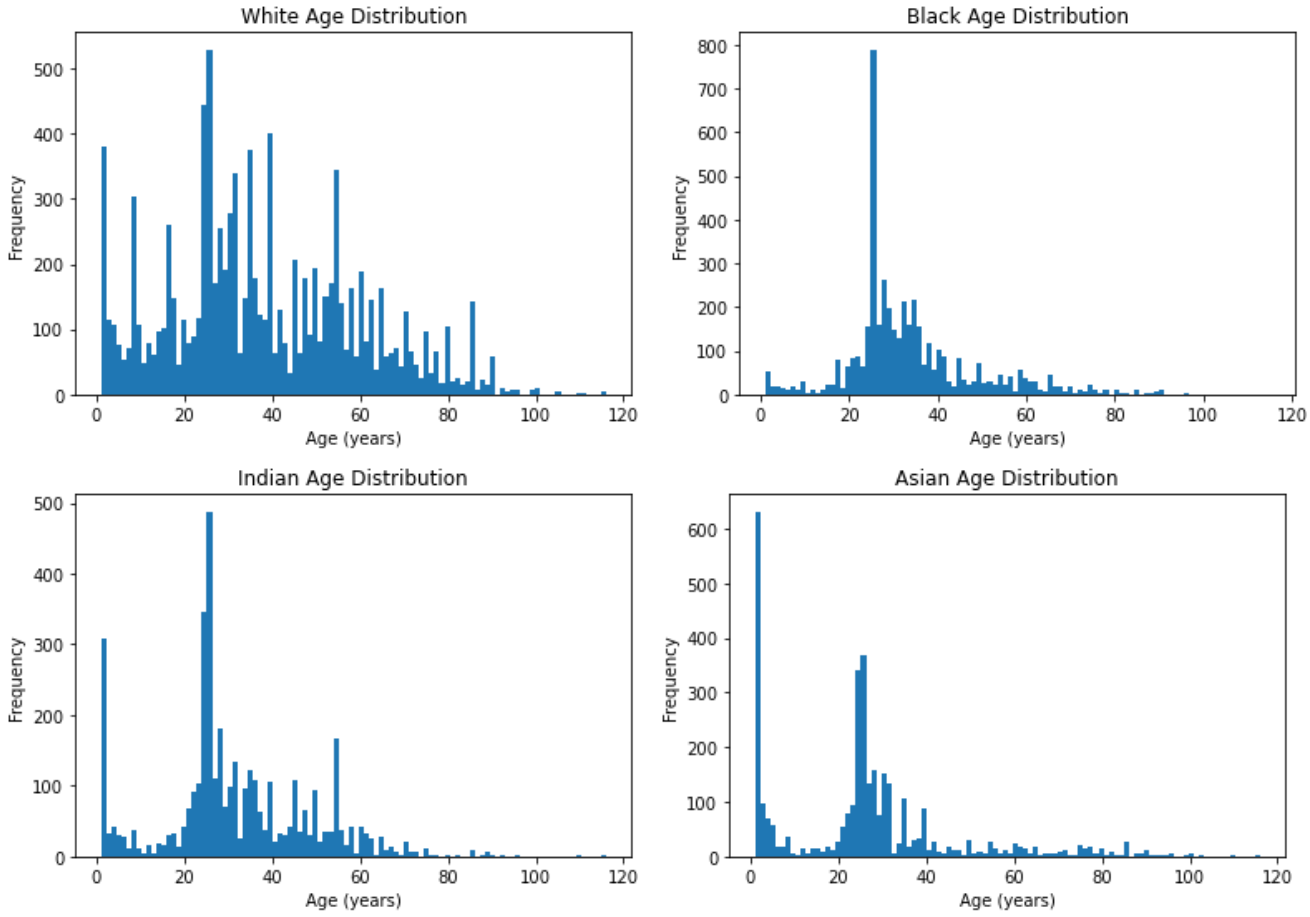

Figure 7 shows the distribution of age for the subset of each of 4 races. Note that each distribution is different, and that it is thus possible that the model learns associations between age and race, when it should not.

Due to memory limitations, dimensionality reduction was performed on the original dataset, scaling all the images down to 100x100, instead of the original 200x200. Facial features were still clearly visible across all images, and this scaling likely equalized the resolution of the images.

Model Selection

In order to achieve the goal of both performing dimensionality reduction, and generating images of faces, a collection of models were used. In particular, an autoencoder was used to create an efficient non-linear latent space, and PCA was deployed in addition to uncover more meaningful features that could be used for face generation.

The first component of the model was an autoencoder, which learned to map 100x100 RGB images of faces to a latent space, and then back to an approximation of the original 100x100 RGB image. The encoder consisted of a series of convolution layers, which reduced spatial dimensions by half, while doubling the number of channels until the spatial dimension was 1x1 (with 256 channels). The decoder consisted of a series of transpose convolutions, which essentially performed the inverse of the convolution layers in the encoder, doubling spatial dimensions while halving channels. The ReLU activation function was used at the output of each layer. The encoder could be used to perform dimensionality reduction on images of human faces

All images from the training set were then passed through the encoder half of this autoencoder, and PCA was applied to the results. Finally, a decoder was trained to map the PCA representation of the faces back to the original 100x100 RGB image of the face. This decoder could be used to generate human faces.

The purpose of the autoencoder was multifaceted. From the perspective of developing the necessary models, there were two goals: first, to explore the necessary-sized latent space to encode a human face without losing accuracy, and second, to create a model that could provide a meaningful latent space. Although an untrained encoder could have theoretically been used to provide the input to the PCA algorithm, it was suspected that a trained encoder (which requires a trained autoencoder) would produce a latent space that could better differentiate between faces. With regards to the goal of the project, the encoder/decoder approach with deep learning allowed for a non-linear mapping to a latent space, which likely allowed for a smaller, yet more complex complex latent space.

However, because the joint distributions of the components in the latent space were unknown, many of the features in the latent space would not have been intuitive or meaningful to a person. Thus, PCA was employed to identify a set of features that would likely result in more variability in the output image, potentially creating more intuitive features from which to generate faces.

Results: Autoencoder and Dimensionality Reduction

In order to determine the optimal size of the latent space to encode an image of a face, autoencoders with different sized latent spaces were trained, and their performance evaluated. Data was partitioned into an 85/15 train-test split. A 5-fold cross validation revealed that there was little variance in performance among folds, so given the increased computational complexity of running cross-fold validation on deep learning models, this train-test split was used for training and evaluating all models.

All autoencoders were trained with learning rate decay, starting with an initial learning rate of 0.0005, that was reduced by 95% every 10 epochs, for a total of 500 epochs. Models were evaluated using binary cross-entropy, a loss function that compares the differences between a multidimensional prediction and label. By minimizing this loss function, the model would be trained to replicate the input image as accurately as possible. As can be seen in **Table 1**, the autoencoder with a 128-dimension latent space converged to the lowest test loss.

Table 1

Latent Space Size	Test Loss Convergence Value
64	0.5580
128	0.5542
256	0.5546

Table 1 shows the binary cross-entropy value that each trained model converged to. The reported values are the mean values of the binary-cross entropy loss in the final 10 epochs of training. Note that the model with an 128D latent space performed slightly better than the 256D latent space model, but much better (relatively) than the 64D latent space model.

Because the test loss is rather abstract, a visual inspection of the autoencoder's output was also performed. **Figure 1** shows the differences in the encoding of a sample face across the three models trained. Note the similarity between the 128D and 256D output, and the slight inaccuracy of the 64D output.

Figure 8

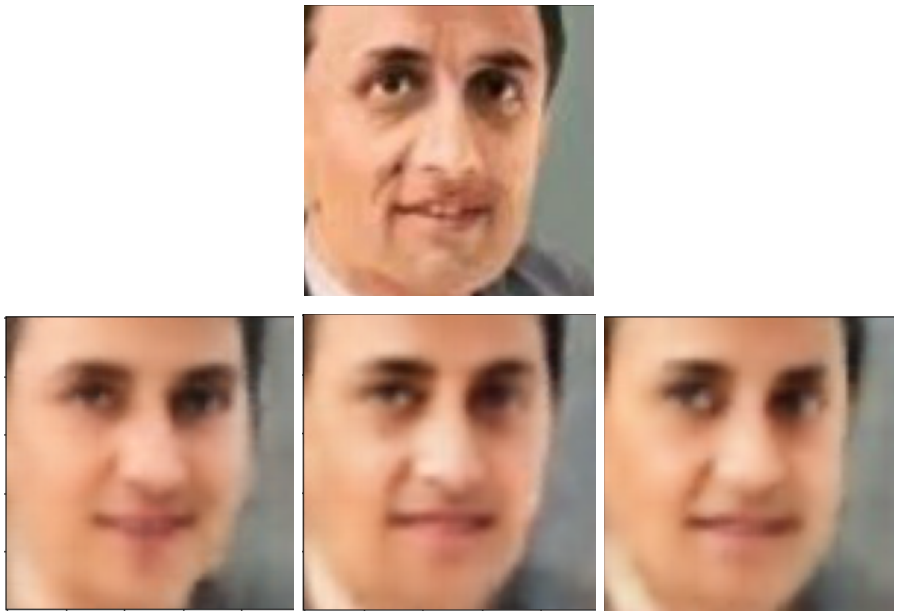

Figure 8 shows, from left to right, the output of the 64D, 128D, and 256D models, for the input face on top. Note that although the outputs of the 128D and 256D models are similar, the output of 64D model seems more inaccurate, failing to model the dark spots under the eyes, and the creases around the nose.

Given that the output of the 128D and 256D model were similar, but better than the 64D model, it was concluded that the 128D model was ideal for the dataset. Although the difference in test loss between the 128D and 256D models may be statistically insignificant, the lack of clear improvement between the 128D and 256D models suggests that 256 dimensions does not significantly improve performance. It can therefore be concluded that about 128 features are needed to encode a 100x100 image of a human face similar to those in the UTKFace dataset. It should be noted that other datasets with different types of faces may require a different number of features.

The performance on these models seems to be the best performance the given architecture is capable of. **Figure 9** shows the training curve for the 128D model. Note that the model clearly converges to a loss of about 0.555.

Figure 9

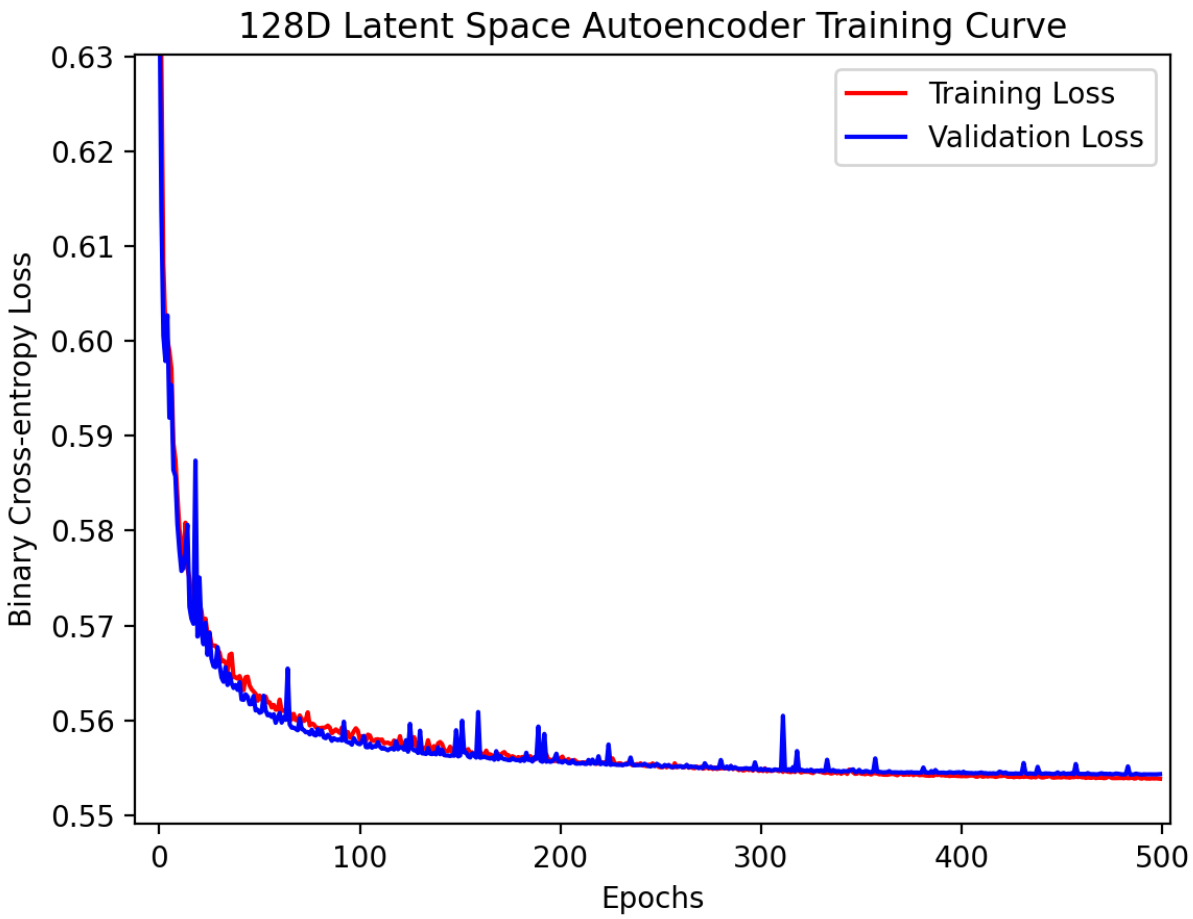

Figure 9 shows the training curve of the 128-dimension latent space model. Note that training and validation loss track each other throughout, suggesting the model generalizes very well. Note also that the loss clearly converges to about 0.555. Although there is no divergence in the validation loss, the training loss does begin drop below the validation loss after about 400 epochs, suggesting the model may be beginning to overfit

---

An audit of a larger sample of autoencoder output was performed to understand the model's behavior. See **Figure 10** for instances where the autoencoder performs well. See **Figure 11** for some instances where the autoencoder performs poorly. In general, the autoencoder is very capable of reconstructing faces, although it does tend to smooth out facial features. This may be because the model lacks a large enough latent space to accurately model features such as wrinkles. It may also be because the dataset is insufficiently sized for the model to recognize that fine details such as wrinkles are not image noise.

Figure 10

Figure 4 shows samples of autoencoder input (top) and autoencoder output (bottom) in cases where the autoencoder performs well. Note that the autoencoder is capable of removing noise and watermakers from photos, as well as modelling facial hair, and eye position (even if not directed at the camera). This degree of reconstruction accuracy was typical for the model.

Figure 11

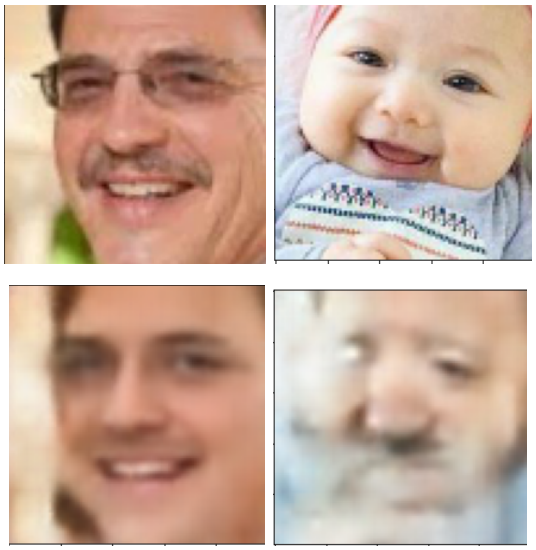

Figure 4 shows samples of autoencoder input (top), and corresponding output (bottom), in situations where the autoencoder struggled. In the example on the left, the model struggles to recreate glasses. In the model on the right, the model fails to reconstruct a face at all, which may be due to the angling of the photo. These types of inaccuracies were generally rare.

The model was also evaluated on subsets of the data, to determine any biases in reconstruction accuracy. See **Table 2** for a comparison of performance across age, **Table 3** for a comparison across gender, and **Table 4** for a comparison across race. Although the differences in model performance on each subset were relatively small, the differences were counterintuitive. In particular, white faces, which represented the greatest proportion of the dataset, were reconstructed least accurately. The differences in performance across each subset seems to be partially due to differences in the mean age of each subset. Older faces were generally reconstructed least accurately, in part because there were fewer samples of older faces than younger faces, and also because older faces tended to have wrinkles and other facial features the autoencoder struggled to model. Other differences in reconstruction accuracy are likely due to differences in image quality across gender or race.

Table 2

Age Range (Years)	Model Performance (Binary cross-entropy)
0-9	0.5543
10-19	0.5497
20-29	0.5366
30-39	0.5517
40-49	0.5662
50-59	0.5753
60-69	0.5767
70+	0.5799

Table 2 shows the performance of the model across different age ranges. Note that older faces tended to be reconstructed less accurately than younger faces. These differences in performance may be due in part to the number of samples in each age group. There were the most samples in the 20-29 age range, and the least samples in the 70+ age range, which correspond to the best and worst performing age groups.

Table 3

Gender	Model Performance (Binary cross-entropy)	Mean Age (Years)
Male	0.5630	35.7
Female	0.5439	30.7

Table 3 shows a comparison of model performance between male and female faces. Note that women's faces were reconstructed more accurately than men's faces. The mean age of the male and female faces is also provided, and given the results in **Table 2**, it is possible that the age differences partially explain the performance difference between genders.

Table 4

Race	Model Performance (Binary cross-entropy)	Mean Age (Years)
White	0.5653	38.0
Black	0.5431	33.9
Asian	0.5359	25.9
Indian	0.5552	31.5
Other	0.5485	23.2

Table 4 shows the performance of the model across different races. Note that although the dataset consisted primarily of white faces, they were reconstructed the least accurately. Given the result in **Table 2**, it is likely that the differences in mean age explain the difference in performance between race, despite the dataset imbalance. This suggests that a dataset with a better balance of age would be important for creating an unbiased model.

Results: PCA and Face Generation

Although the autoencoder proved to be a successful model for performing dimensionality reduction and determining the minimum number of features required to encode an image of a human face, it fails to explain the types of features the model had identified. The latent space lacked meaning because the transformation was non-linear, was not ordered by importance of feature, and had an unknown distribution. **Figure 12** shows the results of trying to create faces by perturbing values in the latent space or generating random latent space vectors.

Figure 12

Figure 12 shows a sample of images generated by the autoencoder by either perturbing values in the latent space, or generating faces from random latent space vectors. Note that although the pictures generally tend to resemble human faces, they do not look realistic. This is because the features in the latent space have an unknown distribution, and are not ordered by importance.

By performing PCA on the output of the encoder, and training a new decoder to map the pca-transformed latent space back to the original face, it was possible to get a better understanding of features that the autoencoder identified. By ordering features by importance, and providing measures of the percent of variance explained, it is possible to create more realistic faces by generating random latent space vectors. See **Figure 13** for a sample of the features that the model identified. The most influential features were generally related to the photo's lighting, including the brightness, as well as the color and direction of the light source. Other influential features modeled facial orientation, as well as gender-related facial structures. Less influential components tended to be more abstract, and it was difficult to determine their purpose.

Figure 13

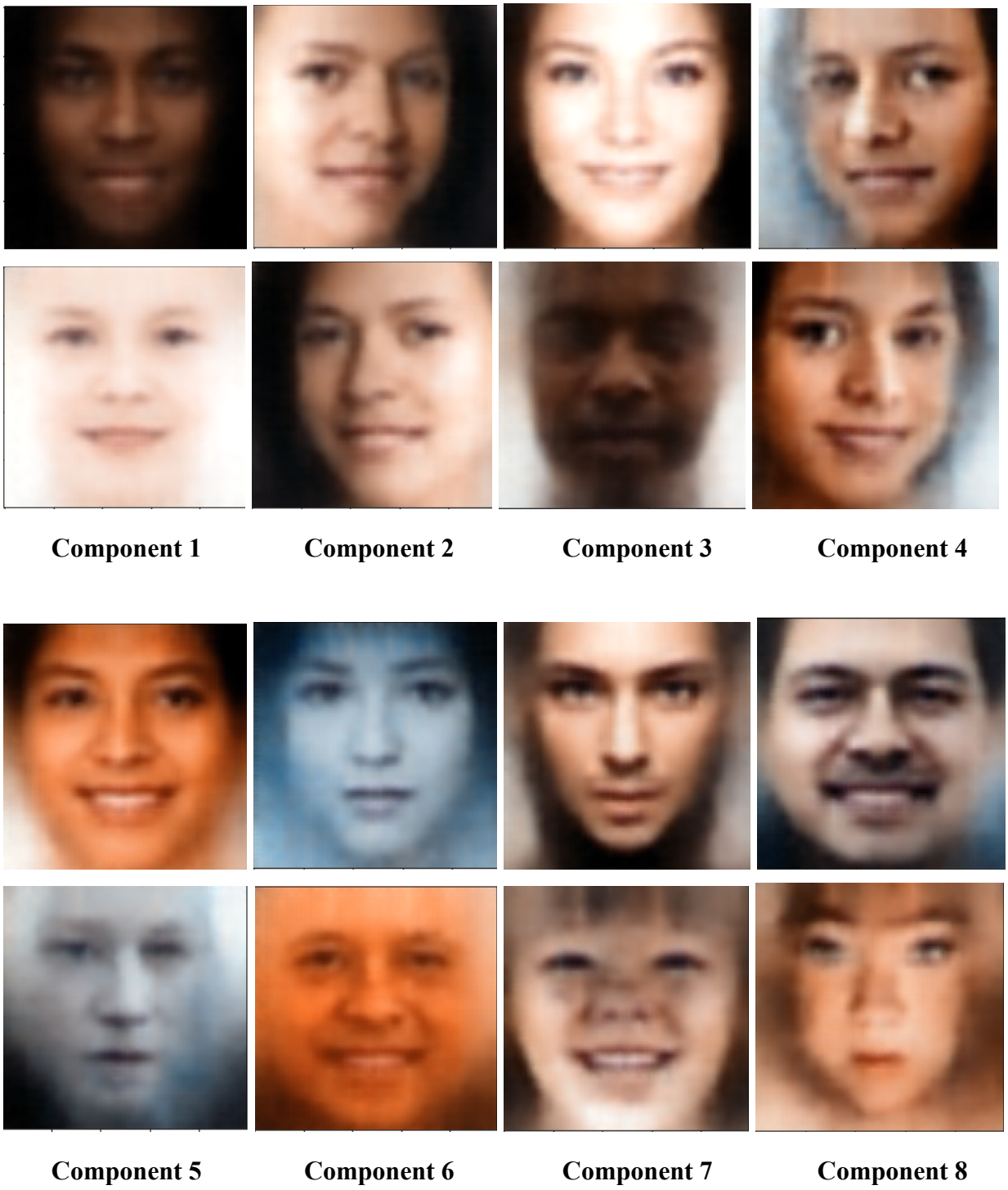

Figure 13 displays the decoder output when setting the values to -1 (top) and +1 (bottom) for each of the first 8 components. Note that many components are related to lighting, while others are related to the orientation and structure of the face. Some, like component 7, seem to model both lighting and facial structure

With meaningful components ordered by importance, it was possible to generate random, realistic human faces. To generate a random latent space vector, gaussian random numbers were generated for each component, with mean 0 and standard deviation 1, and then each component was scaled by the proportion of variance explained that component. See **Figure 14** for a collection of randomly generated faces.

Figure 14

Figure 14 shows a sample of random faces, which were generated by producing random latent space vectors, and decoding them into 100x100 images.

Miscellaneous Results

To gain a deeper understanding of the models and dataset, a variety of other experiments were also performed. First, it should be noted that using a pre trained encoder as input for PCA wasn't necessary. An untrained encoder could have been used, and the decoder could still have learned to map the representation to a human face. However, as seen in **Figure 15**, using an untrained encoder in this scenario reduces the performance of the corresponding decoder, potentially because the pretrained encoder produces a more efficient latent space.

Figure 15

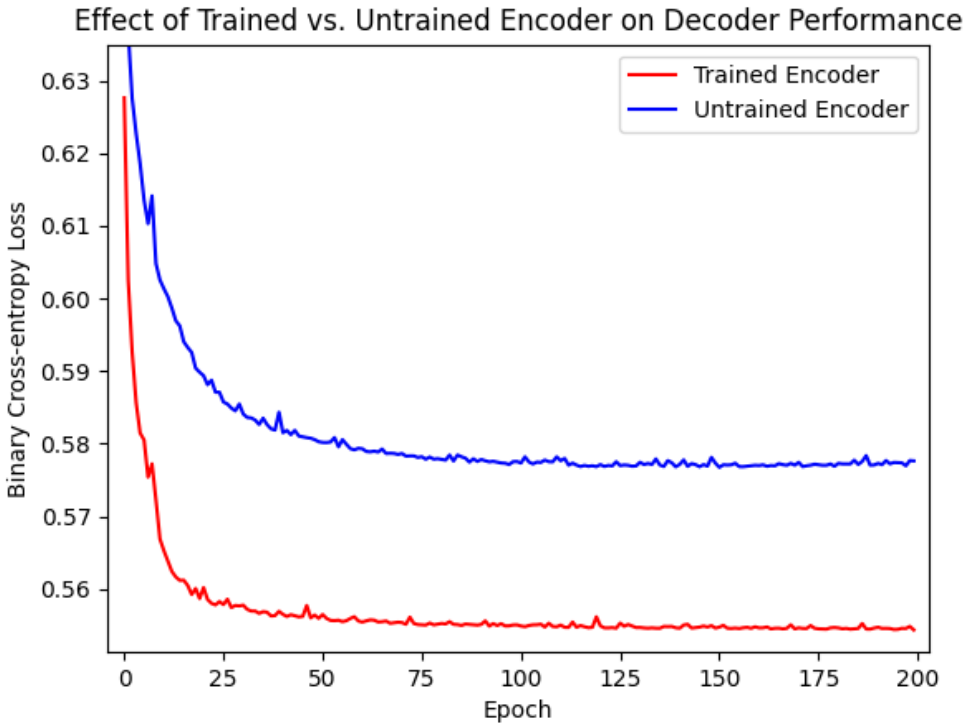

Figure 15 shows a comparison of the training losses for the decoder, given input from a pretrained encoder, and an untrained encoder. Note that pretrained encoder converges to much better performance. This is likely because the output of the untrained encoder is inefficient, limiting the performance of the corresponding decoder.

See **Figure 16** for a learning curve, which demonstrates that the dataset was likely of sufficient size, but that a larger dataset could have marginally improved performance.

Figure 16

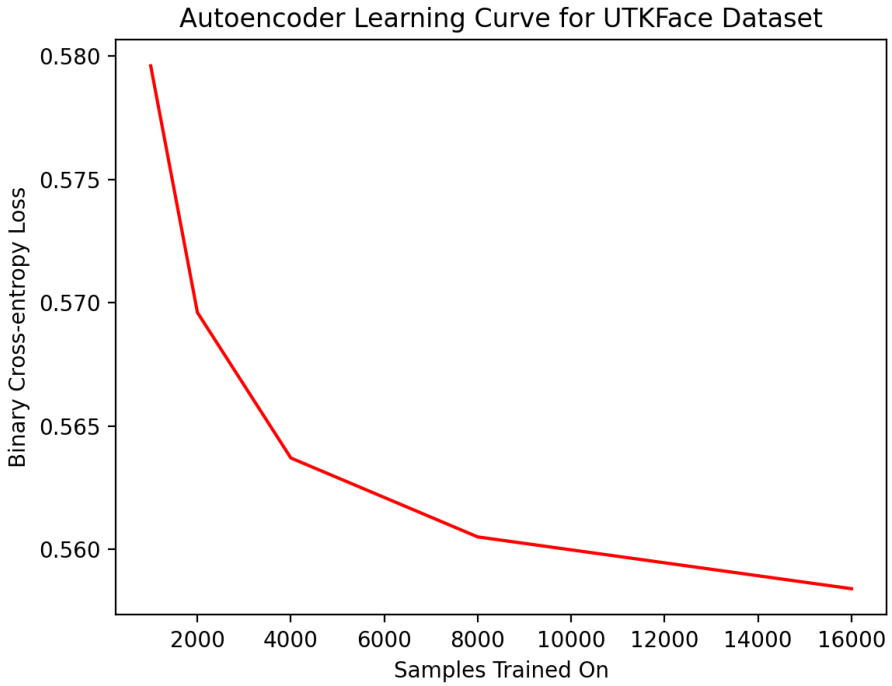

Figure 16 shows a learning curve for the data. Each autoencoder was trained on a total of 1,600,000 steps, although the number of unique samples was changed for each model. Note that it appears that a larger dataset could have marginally improved model performance

The decoder used to generate faces was also used to modify a face. A latent space representation was generated for a face, then individual components of the latent space were modified, and the resulting faces visualized. See **Figure 17** for a sample of the results. Note that perturbing individual components seemed to form faces that appeared to be hybrids of the input face and the component faces, removing some facial features of the input image even for components that are primarily related to lighting

---

Figure 17

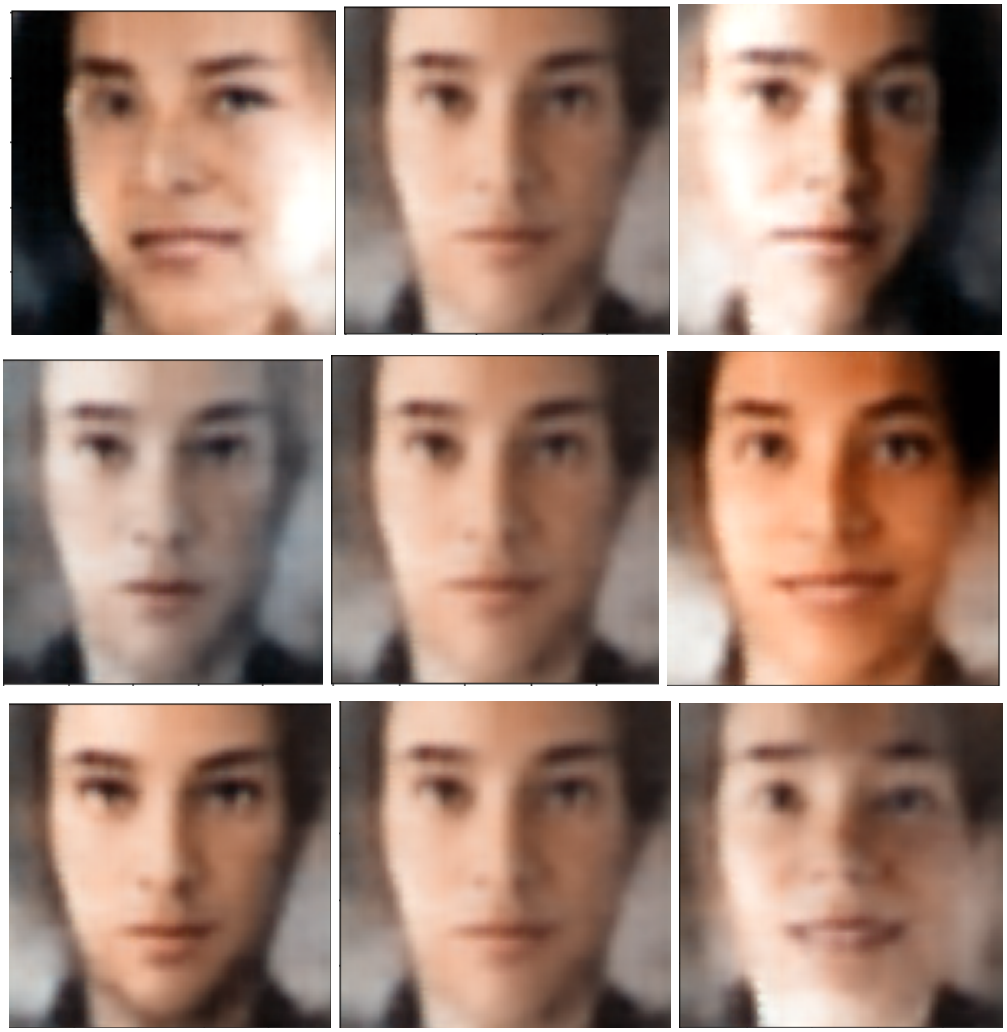

Figure 11 shows the results of perturbing an input face across a few different components. Each row shows the original face in the middle, and modified versions on the left and right, for 3 different components

Finally, an attempt was made to inject an age vector into the latent space, to allow for age progression/regression. Possibly due to the limited variability of age in the dataset, this did not result in any changes to the faces.

Conclusion

In general, this project was successful, not only completing its original research objectives, but also expanding into other topics, like face modification. This project yielded the conclusion that about 128 features are needed to represent a 100x100 image of a human face, and that many of the most important features needed encoded lighting and gender-related facial structure.

The resulting models showed potential to be used in a variety of applications, including image denoising, face generation, and face modification. The autoencoder was moderately accurate at reconstructing human faces, failing to model finer details of the face, but still producing images that were clearly recognizable as human faces.

Through this project, the UTKFace dataset was also analyzed, revealing that its lack of balance in ages may inhibit the development of an unbiased model, and that a larger dataset may improve the performance of the models developed during the course of this project.

In reaching these conclusions, the primary validation metric used was binary cross-entropy, which was used to measure the autoencoder's ability to reconstruct a face from the smaller latent space. The primary validation technique was a train-test split used to verify that the models were generalizing well. A variety of visualizations were also employed to validate the models. These included the training curves to verify the models were converging, but not overfitting, as well as visualizations of autoencoder output, to evaluate and compare model performance in a more concrete way, and verify that the model outputs resembled human faces.

Resources

[1] Original UTK-Face Dataset: https://susanqq.github.io/UTKFace/

[2] Kaggle UTK-Face Dataset https://www.kaggle.com/abhikjha/utk-face-cropped

[3] Code for this project: https://www.kaggle.com/colinsiles/data-science-semester-project