Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Master Thesis

Jan Scholz

Stochastic optimization of synthetic data for neural net based 3d face synthesis

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Jan Scholz

Stochastic optimization of synthetic data for neural net based 3d face synthesis

Master Thesis eingereicht im Rahmen der Master's examination

im Studiengang Master of Science Computer Science
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Andreas Meisel
Zweitgutachter: Prof. Dr. Philipp Jenke

Eingereicht am: 14. Januar 2019

Jan Scholz

Thema der Arbeit

Stochastic optimization of synthetic data for neural net based 3d face synthesis

Stichworte

generative modelling, deep neural networks, generative adversarial network, gaussian mixture model, regression, principal component analysis

Kurzzusammenfassung

In dieser Arbeit wird das Basel Face Model 2017 (BFM) im Hinblick auf die Generierung von Lerndaten für Regressions-Netze untersucht. Ein Regressions-Netz wird erstellt das aus Eingabebildern von Gesichtern Parametervektoren für das BFM erstellt. Diese Parametervektoren sind eine vergleichsweise nierdig dimensionale Repräsentation von Gesichtern die dann in einem nächsten Schritt als Pointcloud oder Mesh bereitgestellt werden können. Das Regressions-Netz wird mit unterschiedlichen Trainingdaten angelernt und die Performance des Netzes und somit die Qualität der Lerndaten wird anschließend mit Hilfe von Facerecognition ausgewertet.

Der Ereignisraum des BFM wird untersucht und mit Hilfe von Generativen Modellen wird versucht den Ereignisraum so zu beschränken, dass bei der Lerndatengenerierung keine ungültigen Gesichter erstellt werden. Die Beschränkung des Ereignisraums auf valide Gesichter gibt den Generativen Modellen die Möglichkeit ausschließlich realistische Gesichter zu generieren. Die Generativen Modelle die in dieser Arbeit erstellt wurden sind eine Auswahl von Gaussian Mixture Modellen und ein Generative Adversarial Network das zudem mit Parametern für das Alter und Geschlecht der Gesichter versorgt werden kann um so weiter die Ausgabe zu beschränken.

Die Wesentlichen Erkenntnise dieser Arbeit sind, dass herkömmliche Erstellung von Lerndaten mit 100k Bildern mittels normalverteilter Werte für die Parametervektoren schlechter abschneidet als die gleichverteilte Initialisierung von Parametervektoren. Die Einschränkung des Ereignisraums des BFMs durch generative Modelle hatte zur Folge, dass zwar realisterische Gesichter erstellt wurden, allerdings wurde so auch die Vielfalt der Daten beeinträchtigt. Der Verlust an Vielfalt kann ein Grund für die etwas schlechter Abschneiden der generativen Modelle sein.

Jan Scholz

Title of the paper

Stochastic optimization of synthetic data for neural net based 3d face synthesis

Keywords

generative modelling, deep neural networks, generative adversarial network, gaussian mixture model, regression, principal component analysis

Abstract

In this work, the Basel Face Model 2017 (BFM) will be examined with regard to the generation of learning data for a regression network. A regression network is created that infers parameter vectors for the BFM from input images of faces. These parameter vectors are a comparably low dimensional representation of faces which can then be provided in a next step as point cloud or mesh. The regression network is trained with different training data and the performance of the network and thus the quality of the learning data is then evaluated using face recognition.

The event space of the BFM is examined and generative models are used to limit the event space such that no invalid faces are created during the generation of the learning data. The limitation of the event space on valid faces gives the Generative models the possibility to generate only realistic faces. The generative models created in this work are a selection of Gaussian Mixture models and a Generative Adversarial Network that can also be fitted with facial age and gender parameters to further restrict the output.

The main findings of this thesis are that conventional generation of 100k images as training data with normal distributed initialization of parameter vectors does worse than uniform distributed values for the parameter vectors. The limitation of the BFM's event space by generative models meant that realistic faces were created, but this also affected the diversity of the data. The loss of diversity can be a reason for the somewhat poorer performance of the generative models.

Contents

List of Tables

List of Figures

1 Introduction

In todays information technology landscape machine learning models play an increasing role in optimizing processes and enabling new technologies. The access to sizable datasets allows for large scale deep neural networks to be trained, but in some areas data is sparse and deep networks can't be generated with the available data.
A possible solution to the sparse data problem is the extrapolation of the existing data. This can be done by data augmentation or with sophisticated generative models. Extensive knowledge of the data is necessary to choose the right model and the right hyperparameters. It is also feasible to solely depend on synthetic data and later use the resulting model in a real world setting. This has been successfully done for example by Richter et al. [1] for semantic labeling of street scenes or more recently with AiFi's[1] checkout free shopping system with similar semantically labeled shopping scenes.

An area with acute sparsity of learning data is 3D representations of faces. Existing databases are small (see Table 1.1) and capturing the data individually is costly and time consuming to the extend that its not feasible for the huge amount that is needed for deep learning tasks. 3D Morphable models (3DMM) are predestined to tackle this challenge, they can compile a vast range of 3D faces and enable large scale generation of training data.

In this thesis a stochastic approach to artificial data generation for a machine learning task will be proposed. The task is to create a 3d representation to a corresponding input image of a face. The output of the machine learning model is a parameter vector which is a condensed representation of a face and can be translated to a 3d model by the Basel Face model 2017 [2]. The objective is to explore the face space of the BFM and create generative models which can sample good instances for similar machine learning tasks. Different methods will be utilized to generate faces and the quality of these faces will be measured by using them as learning data for the mentioned machine learning task. The

[1] http://aifi.io

Name	Samples	Polygons
cyberware.com	100♀100♂	10^6
Basel Face Model DB	100♀100♂	10^6
www.micc.unifi.it/masi/research/ffd	53	$\approx 10^5$
3D-TEC (Twins Expression Challenge)	214	
FRGC v2	≈ 2000	
ND-Collection	27	
MeIn3D	9967	

Table 1.1: Available Datasets

parametric output faces are compared with the original input faces by utilization of face recognition. The similarity scores for the image pairs will then be used as a measure for the quality of the provided learning data.

The main contributions of this thesis are as follows:

- The face space is defined and explored
- A CNN is introduced which is capable to regress BFM parameters from a single image of a face
- A Generative Adversarial Network is proposed which can exploit the BFM face space and sample reasonable BFM instances
- different methods to create synthetic data for the BFM are evaluated and compared

The thesis is structured in six parts. Firstly related work in this area is examined. In the chapter background the main technologies used in this thesis are presented, namely deep neural network, generative models and the Basel Face Model. In the chapter Analysis the BFM event space is defined and possible compositions for the training data are evaluated. Furthermore methods are presented on how to measure the quality of the created synthetic data. The conceptual framework of the thesis is discussed in the chapter Approach. The developed datasets are subsequently evaluated on declared terms and compared. This work is concluded with the most significant discoveries and an outlook to future work.

2 Related Work

An equal approach with synthetic data generation with 3DMM's for facial data has been undertaken in several other works. Two paths were essentially taken, either a random sampling has been done to create faces or a fitting algorithm was applied to images to find realistic faces for the training data.

In [3] synthetic data was generated with parameter vectors which have been sampled from a normal distributed random variable. The generated learning data was used to train a neural net for face recognition. Real data was later introduced in the training to fine-tune the net. The performance of the resulting net supports the thesis that artificial learning data can boost a face recognition net. Suchlike generation of synthetic data was done in [4] and [5].

In Zhu et al. [6] a cascading CNN is created called 3DDFA. In an iterative process the net is fed with the output of the last iteration, thereby refining the result. The output is in the form of a projected image, representing the rendered face. The r,g,b channels are mapping to the normalized coordinates of the model ($r \to x$, $g \to y$, $b \to z$). The iterative refinement of the output has also been done in [4] and resolves the necessity for high quality training data before training.

A fitting approach was done in [7] where multi-view images of the same person from the CASIA WebFace Dataset where used to regress several 3DMM vectors that where later unified to one weighted linear combination, similar to the proposed method in [8]. The resulting training data does enable a neural net to produce consistent output for different images of the same person.

The parameter representation of the BFM can be understand as a facial identity encoding. This encoding is related to the encoding manufactured by face recognition algorithms. Face recognition algorithms pertain to discriminative models and 3DMMs can be categorized as generative models. As both models performances rely on the goodness of the identity representation, the union of both models to create a single new model is therefor conceivable. Recently in Genova et al. [9] Google's FaceNet [10] has been incorporated to regress an identity representation which is forwarded to a 3DMM

regression Network. The ability of FaceNet to predict environmentally invariant identity representations has been exploited to receive cleansed depictions of facial identities. This approach is semi supervised with only part of the training data from synthetic data and the other part from real images which are regressed to 3DMM instances, then rendered and eventually evaluated.

3 Background

3.1 The BFM

The BFM is a 3D Morphable model for faces. It is based on the 3D data of 200 faces split into texture and shape. The BFM spans a 398 dimensional continuous sample space of possible faces of which 199 each are for shape and texture. The BFM is also able to model facial expressions, with 50 parameters the shape of the model can be deformed to form different expressions. All models are based on the same points which are changed in position by the parameters for shape and expression and changed in color by the texture parameters. The proposed models, in this thesis, use only 50 parameters for each texture and shape to simplify the problem with a marginal loss in expressiveness. These dimensions are the product of regression by principal component analysis (PCA), this determines a lesser influence per dimension with descending order. Shape and texture are not linked in this model. The components for all parameter vectors are sorted by variance. The first component of each texture and shape changes the appearance of the model most significantly.

A downside of the BFM is the loss of detail. The information loss is indeed minimized with the principal component analysis but loosing information due to regression to a lower dimensional space is almost inevitable. Smaller details like bumpy skin are not preserved due to the regression. Considering the still staggering expressiveness of this model the loss of minor details is negligible.

3.1.1 Constraints

The BFM can not fully replicate a real human face, the constrains are given by finite computational power and memory, contrary to a continuous reality. The mesh and texture of a model instance can only ever be an approximation of a real face. Another constraint of the BFM is the limited number of samples which it is based on. The full range of appearances of human faces can not be displayed with a small sample size. The samples are also biased for age, over 70% of the faces are between 18 and

30 and only 14% are in the range of 50 to 80 years, for ages higher than 80 no faces are available. The BFM bases its parameters on PCA. The PCA assumes the data to be normal distributed. After the PCA the components are decorrelated, but with not fully normal distributed data the principal components may still statistically dependent on each other. Non linear relationships between features are not considered by the covariance matrix, a good example for this is the Anscombe's quartet [11] (see Figure 3.1) where 4 different datasets have the same correlation but visually differ strongly. All four datasets have the same covariance matrix, the principal components are the eigenvectors of the covariance matrix of a dataset.

$$\Sigma\Gamma = \lambda\Gamma \tag{3.1}$$

$$PCA(X) = \Gamma^T X \tag{3.2}$$

$$\begin{aligned} \text{with } \Sigma &= \text{covariance matrix of dataset} \\ \Gamma &= \text{eigenvectors (principal components)} \\ \lambda &= \text{eigenvalues} \\ PCA(X) &= \text{PCA transform of } X \end{aligned}$$

Therefore all four datasets would have the same principal components.

Egger et al. [12] shows that facial data is not fully normal distributed and this leads to artifacts in face models created with PCA. In Figure 3.2 artifacts, that are the product of not handled dependencies between parameters, are shown. In [12] it is proposed to use a semiparametric gaussian copula model which models dependency and variance independently.

It is also unfavorable that the model is not illumination invariant. In Figure 3.3 the parameters 3 and 4 of the textural component are shown for higher positive and negative emphasis. This will lead to poorer disentanglement of illumination in training data and 3d model instance parameters.

It can also be assumed that the BFM is not invariant to facial expressions. The neutral facial expression of persons differ, it is dependent on the current mood and may also depend on other more persistent factors. It is unlikely that the used 3d scans are completely void of facial expressions. In Figure 3.4 two shape parameters are shown

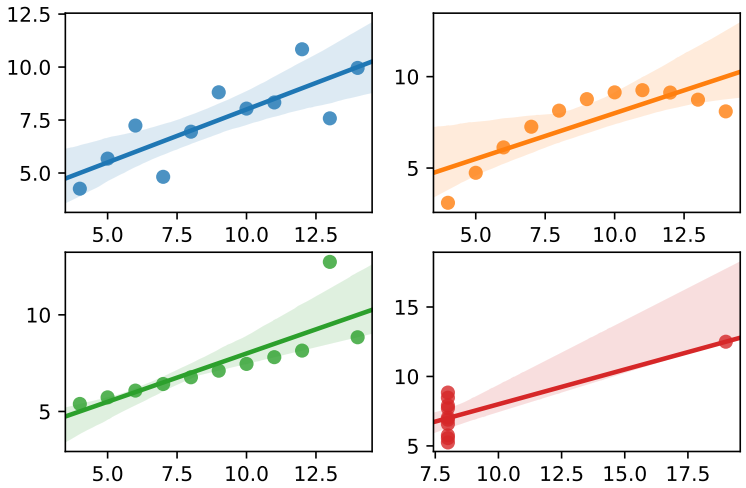


Figure 3.1: Anscomb's quartet, four datasets with nearly exact same mean and variance for x and y and the same correlation and regression line

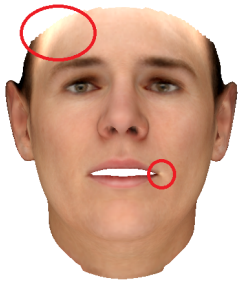


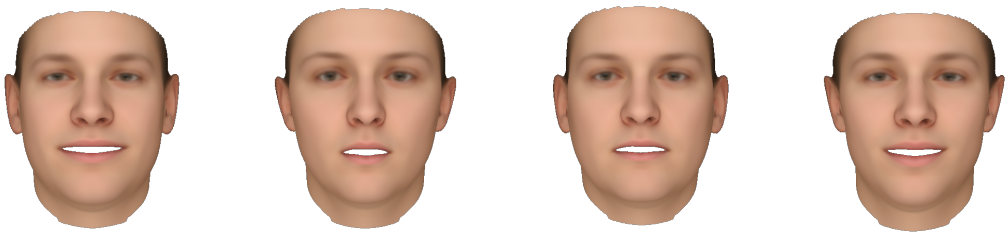
Figure 3.2: Color artifacts

that incorporate facial expressions. One might argue that these expressions are in the spectrum of neutral expressions that occur between identities. But it is indisputable that BFM instances that are rendered with expressions parameters all at zero do differ in their expression.

(a) Texture parameter 3: light coming from top left for positive values and bottom right for negative values

(b) Texture parameter 4: light from top right for positive values and from bottom left for negative values

Figure 3.3: BFM illumination dependency shown for two texture parameters



(a) Shape parameter 10: from smiling to an astounded expression

(b) Shape parameter 15: from a slightly fearful expression to a happy expression

Figure 3.4: Facial expression dependency in two shape parameters

3.2 Machine Learning

The nature of reality can often times be modeled with deterministic models which credibly predict outcomes for certain inputs, i.e. in physics models based on known closed form equations can predict movement of particles or distribution of stress for a building. With growing complexity of systems and lesser knowledge of the underlying mechanics the computability and predictability of models suffers. The approximation of such systems with statistical models reduces computation time and delivers results with sufficient accuracy. These statistical models are developed by optimizing a function for low error for given data and thereby learning the latent function. The formal definition by Mitchell illustrates the *learning* aspect of machine learning.

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."
>
> Mitchell ("Machine learning", 1997)[13]

Models can be separated into discriminative and generative models (see Jebara [14, p. 18]). Discriminative models predict the conditional probability of a given observation.

$$\text{discriminative model: } P(y|x)$$

A generative models on the other hand does compute the joint-probability distribution of x and y

$$\text{generative model: } P(y, x)$$

On the other hand when viewed from the data perspective machine learning can be categorized into supervised learning, unsupervised learning and semi-supervised learning. These concepts either use labeled data, unlabeled data or a mix of unlabeled and labeled data. Supervised learning and unsupervised learning can be further subdivided into discrete and continuous methods (see Table 3.1).

	supervised learning	unsupervised learning
discrete	classification	clustering
continuous	regression	dimensionality reduction

Table 3.1: Machine learning categories for supervised and unsupervised learning

This categorization does however not map generative models which learn the data distribution and infer samples from this distribution. Generative models do distinguish themselves from these categories in the sense that they do not compress information but rather interpolate between given data samples.

3.2.1 Neural networks

A popular variant for function approximation are neural networks, in 1989 Hornik et al. [15] proved that a neural net with one hidden layer and a sigmoid activation function is able to approximate any function. While this is possible there is no guarantee for efficiency and it can be computationally reasonable to add hidden layers to create more complex functions as proposed in [16]. Neural nets consist of neurons (also called perceptron) which are comprised of a bias (b), weights ($\boldsymbol{w}$) for every input ($\boldsymbol{x}$) and usually an activation function (a). Equation 3.3 shows a neuron with leaky relu activation function. The expression $x_{n+1}^{(k)}$ denotes the k-th neuron in the n+1-th layer. Every neuron computes the result of all inputs, in the form of preceding neurons or first level inputs, multiplied with the weights and summed up together with the bias.

$$a_{lr}(x) = \begin{cases} x & \text{if } x > 0 \\ x \cdot \alpha & \text{else} \end{cases}$$
$$x_{n+1}^{(k)} = a_{lr}^{(k)}(\boldsymbol{x}_n^{(k)T} \boldsymbol{w}_{n+1}^{(k)} + b_{n+1}^{(k)}) \tag{3.3}$$

cf. [17]

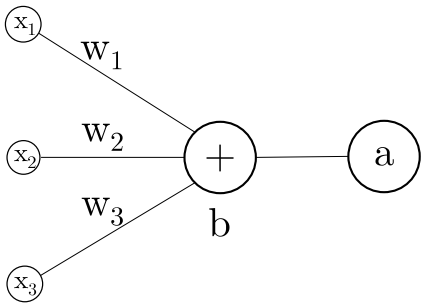


Figure 3.5: A neuron

Neurons are connected in a net with other neurons and learn the desired function. Learning in this case happens by minimizing an objective function (f_o). The result of the objective function is an error value which in the case of a generic feedforward network represents the quality of the guess made by the neural net.

The minimum of a function with an arbitrary number of variables can be approached with *gradient decent* if the functions is differentiable. By computing the minimum of the objective function the latent function which is desired will be approximated. A net can be represented by the simple equation (cf. [18, p. 168]).

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = \hat{\boldsymbol{y}} \tag{3.4}$$

with

$$\boldsymbol{\theta} = \{w, b \mid w, b \in M\}$$
$$w = \text{weights}$$
$$b = \text{biases}$$
$$M = \text{the model}$$

The net f has the parameters $\boldsymbol{\theta}$ and takes the inputs $\boldsymbol{x}$ to compute the result $\hat{\boldsymbol{y}}$. Every layer of the neural net can be represented as a singular function (cf. [18, p. 168]).

$$f_n(\boldsymbol{\theta_n}; ...f_4(\boldsymbol{\theta_4}; f_3(\boldsymbol{\theta_3}; f_2(\boldsymbol{\theta_2}; f_1(\boldsymbol{\theta_1}; \boldsymbol{x}))))...) = \hat{\boldsymbol{y}}$$

For supervised learning the objective function may be defined as the mean squared error, when comparing $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$.

$$f_o = \frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2$$

uron

By feeding values to a network and continuously measuring the error, the gradient of the objective function with respect to the inputs x can be computed.

$$\bigtriangledown_x f_o(f, \boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{y})$$

By moving in the direction of the negative gradient we continuously decrease the value of the objective function. This method is coined gradient descent [19]. And then be propagated from the back of the network, layer per layer, till the front while adapting weights and biases. This algorithm is coined *backpropagation*. The learning process follows the negative gradient of the error function and lowers steadily the error value. The size of the steps ($\Delta\boldsymbol{\theta}$) taken while descending the error function f_o are defined by the learning rate. The change of the parameters $\boldsymbol{\theta}$ in one learning step is determined by the magnitude of the learning rate η (cf. [18, p. 85]).

$$\Delta\boldsymbol{\theta} = -\eta \bigtriangledown f_o(\boldsymbol{\theta})$$

A change of a single weight is determined by:

$$w_i + \Delta w_i \rightarrow w_i$$

with

$$\Delta w_i = -\eta \frac{\delta f_o}{\delta w_i}$$

3.2.2 Convolutional Neural Networks

Convolutional Neural Networks (LeCun et al. [20, 21]) exploit a property of images to compute high dimensional data in the form of images more efficiently. The property is that for every given pixel in an image the surrounding pixels are dependent on this pixel. To visualize this, when viewing an image of a face the pixel on the tip of the nose is highly dependent, with respect to color, on the pixels next to it. A Pixel in the top left of the same image does instead have very little influence on the pixel of the nose and vice versa.

A CNN does a convolution operation with a square shaped kernel (K, also called filter), usually in the size of 3x3 or larger. This kernel will move over the image with a predefined step size in horizontal (h_{step}) and vertical (v_{step}) direction and creates a new matrix called feature map (F). In every step the kernel will be multiplied elementwise and the resulting matrix will be summed up. The number of steps possible in vertical and horizontal direction on the input matrix determines the size of the resulting matrix.

$$F_{k,l} = \sum_{j=1} \sum_{i=1} K_{i,j} \cdot I_{k \cdot h_{step}+i, l \cdot v_{step} j} \tag{3.5}$$

After a convolution an activation function is applied. And between successive convolutions a pooling layer is usually inserted. This pooling layer has several advantages. It reduces complexity while maintaining information and makes features more robust by achieving partial spatial invariance [18, p.342]. Because not all information is retained overfitting is reduced. Scherer et al. [22] show that non overlapping max pooling speeds up convergence and also reduces error. Pooling is applied likewise the convolution on a square section of the input, moving over the input feature map while only moving once over every cell of it. For every iteration the result for a given pooling operation is computed. The commonly used operations are AVG, SUM or MAX.

3.2.3 Generative Adversarial Network

A special case of generative model is the Generative Adversarial Network (GAN) from Goodfellow [43], which pertains to the unsupervised machine learning algorithms. It consists of two models of which one is trying generate output similar to the provided training data, the generator. And one model which tries to distinguish the fake from the real data, the discriminator. By playing an adversarial game these two models learn together. This delicate relationship is fragile and can be undermined if one party overwhelms the other so that none learn the true data distribution.

This game is essentially a minimax game, in which every player wants to minimize the opponents maximal gain. The value function $V(D, G)$ is depicted in equation 3.6. The generator (G) tries to minimize by maximizing the term $D(G(z))$ and the discriminator tries to maximize $D(G(z))$ by identifying fakes and maximizing $\log D(X)$ by identifying real data. The input for D is denoted with x and has the probability distribution $p_{data}(x)$ for the training data. The generator receives a noise input z which is usually sampled from a uniform distribution $p_z(z)$.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.6}$$

with

$$\begin{aligned} p_{data} &= \text{probability distribution for given data} \\ p_z &= \text{probability distribution for } z \\ \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] &= \text{Expectation for } \log D(x) \text{with respect to } p_{data}(x) \end{aligned}$$

The generator learns the distribution p_g which has its global optimum when $p_g = p_{data}$. The optimum is reach when the discriminator can not differentiate between samples of the generator and real data. When $p_g \approx p_{data}$ the expected prediction of $D(x)$ reaches $\frac{1}{2}$, as can be seen in Equation 3.7. When comparing real data versus fake both will have a mean probability of 50%.

$$D(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} = \frac{1}{2} \tag{3.7}$$

covariance type	shape	expressiveness
spherical	$\mathbb{R}^c$	single variance per component, each component is spherical
diagonal	$\mathbb{R}^{c \times f}$	diagonal Σ per component components are either vertically or horizontally warped between features
tied	$\mathbb{R}^{f \times f}$	every component has the same general Σ, every component has the same shape
full	$\mathbb{R}^{c \times f \times f}$	every component has its own general Σ, every component has its own shape

Table 3.2: Covariance types for the Gaussian Mixture Model. Shape dimensions with: features (f) and components (c)

3.2.4 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) assumes a multinomial Gaussian distribution for a given dataset. The framework that is used in this project for creating GMMs is scikit-learn [23]. The provided model can be served with 4 different options for the covariance matrix. The four options are *spherical*, *diagonal*, *tied* and *full*. The expressiveness of these options increases in the same order (as seen in Table 3.2). The pdf of the GMM is computed analogous to the known gaussian distribution like shown in Equation 3.8 and the following.

$$p(\vec{x}) = \sum_{i=1}^{K} \phi_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) \tag{3.8}$$

$$\mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K |\Sigma_i|}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu}_i)^{\mathrm{T}} {\Sigma_i}^{-1} (\vec{x} - \vec{\mu}_i)\right) \tag{3.9}$$

$$\sum_{i=1}^{K} \phi_i = 1 \tag{3.10}$$

with

Σ_i = covariance matrices

ϕ = component weights

K = number of components

To find the right component size for the GMM two metrics can be examined, the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) (see Equation 3.13. These metrics define a measure for goodness of fit and are both based on the maximum Likelihood. While BIC penalizes heavier with more data available, with the term $\ln(n)$, the AIC does not. The Likelihood function $\hat{L}$ is defined as the sum of all probabilities for every observation under a given distribution, which means for our case under multiple distributions. When the Likelihood function is optimized for maximum value it is ensured that the parameters $\hat{\theta}$ for the model M are also optimized for the data. The underlying algorithm to optimize Likelihood is Expectation maximization. It is guaranteed to progress in every step and, at least, converge to a local maximum or a saddle point.

$$\text{AIC} = 2k - 2\ln(\hat{L}) \tag{3.11}$$

$$\text{BIC} = \ln(n)k - 2\ln(\hat{L}) \tag{3.12}$$

$$\hat{L} = p(x|\hat{\theta}, M) \tag{3.13}$$

3.3 Dimensionality reduction

3.3.1 Principal Component Analysis (PCA)

The PCA [24] is a multivariate analysis method used to condense information for easier data exploration or feature extraction. A n-dimensional multivariate dataset can be reduced to a desired lower dimensional space with [1,n] components. These components do represent concepts inherited in the dataset. Intuitively PCA can be understood as a reorganization of the axes of a multidimensional dataset with maximized variance for each new axis. PCA can be computed based on singular value decomposition [25] which splits a given matrix in it's singular values and singular vectors. This method is computationally less expensive than the already mentioned eigenvalue decomposition of the covariance matrix and thus to be preferred. The singular vectors and singular values are defined according to [25] as

$$\begin{aligned} AV &= DU \\ A^H U &= DV \\ A &= UDV^H \end{aligned} \tag{3.14}$$

with

$$\begin{aligned} A^{n\times m} &= \text{Matrix with } n \text{ samples and } m \text{ features} \\ A^H &= \text{ Hermitian transpose, if } A \text{ is real } A^H = A^T \\ V^{m\times m}, U^{n\times n} &= \text{ normalized singular vectors of A} \\ D &= \text{ singular values of A} \end{aligned}$$

The Matrix V is ordered according to the strength of the singular value of every component, beginning with the highest value. Data can be transformed from feature space (X) to component space (Y) with:

$$X \to Y = XV \tag{3.15}$$

The decision of how many components are used depends on how much of the information is preserved in a given number of components. For each component the explained variance indicates how much information is contained in a component. If the decision has been made to use as much components as necessary to conserve at least 95% of the information, by summing up the normed variances by starting from the first component and descending till a threshold of 0.95 is exceeded we receive the number of needed components.

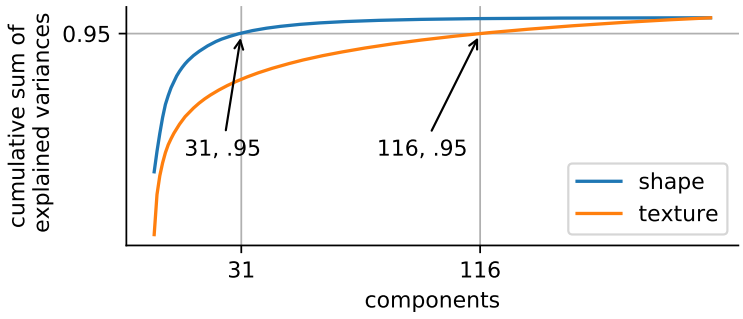


Figure 3.6: Cumulative sum of explained variances per PCA component of the BFM

Looking at Figure 3.6 we see the explained variances for the BFM. The first 31 components of the shape dimensions already explain 95% of the information the BFM inherits. Whereas the color information is only preserved if we take the first 116 components. The data is however not complete and only the first 199 components for shape and color are provided with corresponding explained variance for the BFM. The variances for components over 199 is low but the PCA transforms data to 160.000 values (implying 160.000 principal components), so the number of components needed for %95 of all information is higher. The variance of the ordered components logarithmically nestles to the sum of all variances, due to this it is reasonable to estimate that over 90% to 95% of all information is inherited in the respective 199 components.

3.4 Goodness of fit

3.4.1 Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test [26] is a useful tool to compare data samples with a given distribution. To asses if the dataset comes from the given distribution the Kolmogorov statistic (see Equation 3.16 for discrete and continuous case) can be computed. The ks statistic measures the difference between the distribution (F) which is to be tested and the empirical cumulative distribution of the data samples y. The empirical cumulative function E_N (see Equation 3.17) can be understand as a step function with ordered values of the sample data, iterated from lowest to highest value with an $\frac{1}{n}$ increase when an observation occurs.

$$D = \max_{1 \leq i \leq N} \left(F(y_i) - \frac{i-1}{N}, \frac{i}{N} - F(y_i) \right) \tag{3.16a}$$

$$D = \sup_x \left| F(x) - E_N(x) \right| \tag{3.16b}$$

$$E_N(x) = \frac{1}{N} \sum_{i=0}^{N} [y_i < x] \tag{3.17}$$

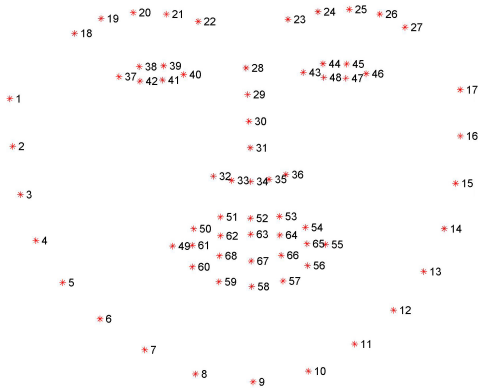
We can reject the null hypotheses that the samples come from the same distribution if the ks statistic is large and the p-value is small. A low p-value indicates that the observation is unlikely to occur randomly. For the two sample ks test the two datasets

are converted to two empirical distribution functions, the computation will then be done with Equation 3.18.

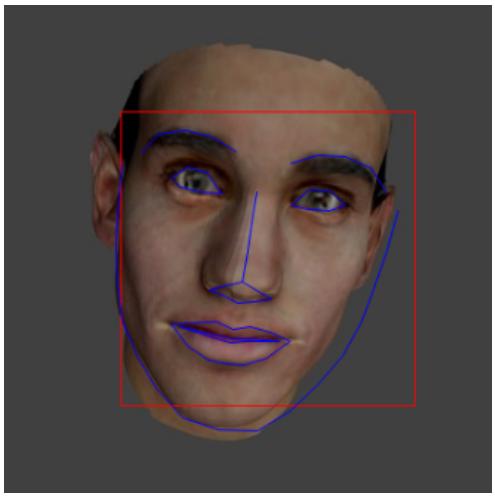
$$D = \sup_x \left| E_M(x) - E_N(x) \right| \tag{3.18}$$

3.5 Face recognition

The complex task of face recognition has become greatly sophisticated over the last 10 years with the dawn of deep neural networks. A lot of other techniques do exist which do not rely on neural nets but can't compete with this technique.
Face recognition is usually coupled with previous face detection. Face detection delivers a position in a given picture with height and width where the face is located. The Dlib [27] library for example does provide face detection based on Haar cascade from Viola & Jones [28] which was published already in 2001 and has since been the industry standard for face detection in cameras. Dlib also extracts 68 facial landmarks, which can be seen in Figure 3.7.
When the face is located and facial landmarks are found these information can be turned over to the face recognition. All face recognition algorithms evaluate the face proposed a vector of metrics which can then be compared the vectors of other faces. The measurement which is used to compare the similarity of the metrics can differ but is generally a distance measure. When an explicit algorithm is created for such a task the used metrics are usually defined as ratios of different segments of the face or distances between points.
The metrics inferred by CNNs are unknown in their manifestation in the face and are obtained by optimizing for greater distance between different faces and smaller distance between similar faces. These metrics, though unknown, work well so that in the last years face recognition surpassed the capabilities of humans. In 2007 it was assessed by Adler & Schuckers [29] that then current algorithms were already better than 50% of humans. In 2014 Phillips & O'Toole [30] concluded that face recognition algorithms outperform human capabilities in all circumstances except for extremely difficult conditions were non-face identity cues are dominant and the pose is not frontal.



(a) Image and Landmarks from Sagonas et al. [31]



(b) The red bounding box shows the result of the face detection and the blue lines connect the 68 facial landmarks found in the image.

Figure 3.7: Facial landmarks and face detection from Dlib.

4 Analysis

4.1 Defining The BFM event space

A relation between a BFM instance and a real face will be defined as a pair with minimized distance for shape and texture. The mapping can be established with a fitting algorithm. For real faces the full range of human appearances can be considered. When an ideal fitting algorithm f is assumed real instances can be mapped as shown below ($\mathbb{R}$: set of real faces, $\mathbb{F}$: set of fitted BFM model instances).

$$f : \mathbb{R} \to \mathbb{F} \tag{4.1}$$

$$\mathbb{A}_i = \mathbb{A} \setminus \mathbb{F} \tag{4.2a}$$

$$\mathbb{F}_v \subset \mathbb{F} \tag{4.2b}$$

$$\mathbb{F}_v = \{a \in \mathbb{A} \mid r \xrightarrow{f} a \ \wedge \Delta(r, a) < \epsilon\} \tag{4.2c}$$

The event space of the BFM ($\mathbb{A}$) consists of invalid and valid instances, the valid instances are part of the set $\mathbb{F}$. Invalid instances ($\mathbb{A}_i$) can be generated by the BFM but do not have a mapping to a real face. Invalid instances of $\mathbb{F}$ do have a corresponding real face but have a to large distance due to the constrained expressiveness of the BFM. This prevents a complete mapping of real faces to the BFM. Therefore an unattainable set of faces does exist which translate to $\mathbb{F}_i$. These faces can't be depicted by the BFM while the size varies with the allowed distance ϵ. The objective is to create a model which generates instances solely of the set $\mathbb{F}_v$.

The vastness of the event space can be illustrated by a simple example. If we separate every parameter into 2 quadrants (positive / negative values) we receive 2^{100} or $1.27 \cdot 10^{30}$ distinct quadrants. This number is so large that we can't even create a single example per quadrant. And even when two instances are inside the same quadrant they can look

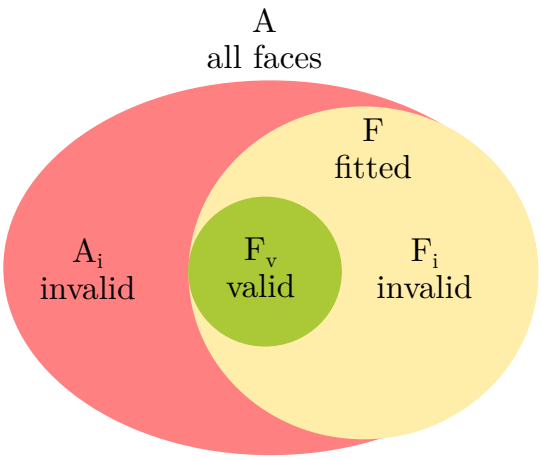


Figure 4.1: Venn diagram of possible faces in the event space of real faces and BFM faces

very different. A brute force solution to find valid faces can therefore be discarded.

Faces near the center of the face space (distance to the parameter vector with all zeros) are usually valid. They may look very average but it is imaginable that a real face exists that is approximated by these instances. The further away a face is from the center the greater the chance that it is not valid. This can be easily demonstrated by comparing different instances with their corresponding standard deviations. The higher the standard deviation the more excessive the features of the face. If we find a valid face in an arbitrary position in the event space the near surrounding of the face can also occupy other valid faces. With minor changes to the face a similar face can be generated which may differ by eye color or look slightly more aged. In conclusion to these statements two conjectures can be made:

1. With increasing distance to the center the probability that instances are valid decreases

2. A valid instance should have valid neighbors in near proximity

If these two conjectures stay true it follows that a generative model, build on valid faces, should be able to sample more valid faces. If valid faces with a greater standard deviation are observed by the generative model it should also be able to sample more valid instances further away from the center. Therefore generative models should be able to sample more distinct valid instances.

4.2 Creating Data

Learning data is provided as images for the input and BFM parameter vectors as labels. Essential for synthetic data is a good similarity to real data which is to be modeled. This similarity for facial images can be viewed on 3 different levels: environment, instance and on a meta level.

Environment level

On the environment level the appearance of the image should resemble a real scene of a photo taken from a face. This is established with several methods, first a background should be put behind the face. This background shall be comparable to a background which is found on real images. Lighting of the 3D face is not only vital for visual similarity, but also for the extraction of the surface condition. More prominent shape features in the face may cast shadows which further support the model with clues for the shape of the face. The pose of the face in the generated images should also be varied in all 3 axes. Images with intense rotations are discarded when no face is found in the preprocessing step.

Instance level

When images are created with the previous mentioned enhancements the resulting images do look similar to real images. If all parameter vectors are randomly sampled the instances are structureless and features are partly random, but real faces have relations between features. The most prominent example for this conjecture is the distribution of male and female facial features. Generic random faces mix female and male features and form a unimodal distribution of male to female looking faces, as can be seen in Figure 4.2. There is a large variety of relations that can be found in faces. An example for a correlation is a larger nose and ears, which do grow with age and therefore also correlate with wrinkles and lighter hair. Women wear makeup and different ethnicities have characteristic textural and shape combinations.

The BFM can express some of these faces but a purely random approach does only partially maintain these relations. The principal components considered each on their own do maintain these relations, but the mixture of all parameters breaks these relations. This applies specifically for feature correlations between texture and shape, because separate principal components are used which have no connection. A face with a feminine

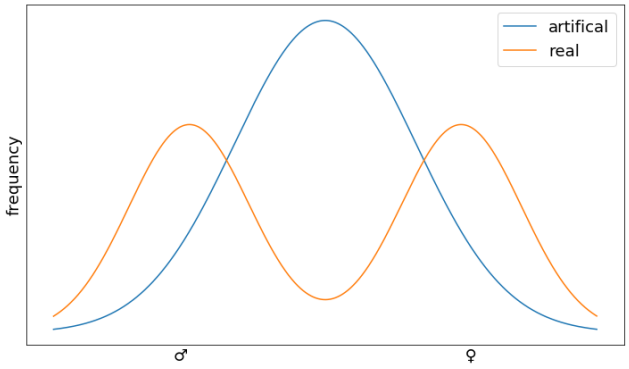


Figure 4.2: An estimate for the distribution of male and female features present in purely randomly generated BFM instances and real faces

shape could have beard stubbles due to randomly sampled texture. To sample good faces these relations have to be considered.

Meta level

When all images are statistically compared to real images differences still emerge. Depending on the later application of the data the composition regarding age, sex and ethnicity should also be considered. This applies particularly to the application of face recognition, where the population in the learning data should be analogous the population that will be examined afterward. In Klare et al. [32] the influence of the demography of the training data for the quality of the predictions is demonstrated.

4.2.1 Sampling

Three different methods to supply image/parameter vector pairs are presented.

Regressive method

Artificial data can be generated by fitting the BFM to images. The image can be used as input and the resulting BFM instance represented as parameter-vectors can be used as label. This method has the advantage that the data is partly real. The resulting data does also preserve correlations between features of the faces. On the downside the label is only an approximation of the real face and has an unspecified margin of

error. A technical difficulty is that fitting the images with an iterative algorithm is computationally expensive and may take between 1-5h per image to get reasonable results. Considering that the needed data is in the range of 10^5 and 10^6 samples, the computation time may be as large as $5 \cdot 10^6$h. This drawback has also been recognized in [7].

Progressive method

A much cheaper method is to sample parameter vectors randomly and synthesis images from these instances. The sampling of the vectors can be done with a normal distribution. The resulting BFM instances do resemble human faces but break existing correlation between facial features. This can be seen most prominently when looking at a selection of generated images of these instances. It does stand out that a large proportion of the faces do not look male or female but inherit features of both sexes, as illustrated in Figure 4.2, some correlations of real faces are evidently no more present. This is true for all correlations which are not inherently tied to a single BFM principal component.

Stochastically optimized method

In this proposed method parameter vectors are obtained by fitting images and using these BFM instances to create a generative models from which an arbitrary amount of instances can be sampled. Images are created from the instances similar to the previous method. With a proper generative model feature correlations in faces should be preserved. Furthermore a generative model which can synthesize instances for given meta variables like age and gender can promote greater diversity and allows for a delicate optimization on the meta level.

The datasets in this thesis are created with the progressive and the stochastically optimized method.

attribute \ method	progressive	regressive	stochastic
coupling image ⇄ model[1]	upper barrier: fitting algorithm	ideal	ideal
authenticity of input image[2]	ideal	average	average
feature correlations[3]	mostly preserved	marginally preserved	mostly preserved

[1] prospect of transformation from input to output and reverse

[2] how real does the input image look

[3] are feature correlations present, i.e. multiple manifestations of female facial features for female persons

Table 4.1: Comparison of methods to create artificial data

4.3 Measuring dataset quality

To correctly compare all sampling methods the architecture and all learning parameters for the CNN are fixed. While this may not the yield the best outcome for every sampled dataset comparability is ensured. To assure significant results every dataset was used 4 times for training to get an accurate assessment of every datasets performance.

method	measurement	applied to
Clustering & Distances	data distribution	training data, predicted data
dimensionality reduction	concept discovery, data distribution	training data, predicted data
Kolmogorov-Smirnov test	goodness of fit	training data, GMM & GAN
similarity score	neural net / dataset performance	predicted data

Table 4.2: Evaluation methods

Clustering

Clustering can be exploited to measure structure by computing various numbers of clusters and measuring the distance from each centroid to each member of the cluster. When many instances cluster together the data is less random and more structured. With the k-means algorithm [33] all created training datasets are evaluated for 2 to 200

clusters and the distances are recorded. This measure for k-means is termed distortion and is computed as shown in Equation 4.3. To compare the results every observation of every dataset is divided by its standard deviation to achieve unit variance ($\sigma = 1$). This scale-invariant technique however does not account for the space a data distribution occupies.

$$\mathcal{L}(\Delta) = \sum_{k=1}^{K} \sum_{i \in C_K} ||\boldsymbol{x}_i - \boldsymbol{\mu}_k||^2 \tag{4.3}$$

Distances

Distance measures will also be applied to the datasets and the predicted BFM instances to show how diverse the data is. The employed measures are cosine- and euclidean distance. The cosine distance does measure the angle between observations while the euclidean distance measures the absolute distance between observations.

The cosine distance is a suitable measure to compare BFM instances. We define the cosine distance as

$$\cos_\Delta(\boldsymbol{A}, \boldsymbol{B}) = 1 - \cos(\boldsymbol{A}, \boldsymbol{B}) \tag{4.4}$$

with

$$\cos(\boldsymbol{A}, \boldsymbol{B}) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{\|\boldsymbol{A}\|\|\boldsymbol{B}\|} \tag{4.5}$$

When the cosine distance is formulated as such the value minimizes for equal angled vectors and maximizes for antisymmetric vectors.

A value of 2 expresses full dissimilarity having one measure on the complete opposite side of the other. A measure of 0 describes full similarity and a measure of 1 means orthogonal alignment of the observations. Cosine and euclidean distances are shown for 3 models in Figure 4.4 where the cosine distance $\overline{ab}$ is 0 and the distance $\overline{ac}$ is 2 the euclidean distance is for $\overline{ab}$ and $\overline{ac}$ are both equally 6.32. The faces a and b look more similar than a and c even though the euclidean distance is the same. The face b has the same features as face a, but they are more pronounced. A small measure for the euclidean distance may still indicate a more similar face than two faces with large euclidean distance and a very low cosine distance. The drawbacks of both distances are clear, cosine distance judges a and b as the same face while euclidean distance estimates

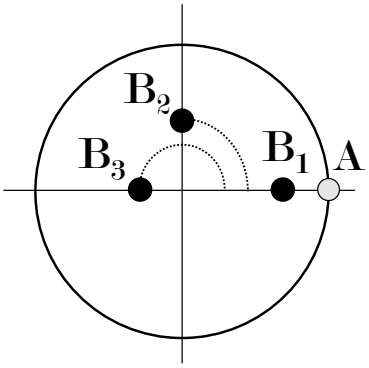


Figure 4.3: Cosine distance visualized for 2 dimensions. $\overline{B_1A} = 0, \overline{B_2A} = 1, \overline{B_3A} = 2$. The scale of the values is ignored only the angle between observations is taken into account.

that a,c and a,b are equally different. Looking at both measures is reasonable to evaluate the data.

(a) 1 σ (b) 3 σ (c) -1 σ

	$\overline{ab}$	$\overline{ac}$
cosine	0	2
euclidean	6.32	6.32

Figure 4.4: All three instances have the first 5 parameters for shape and texture changed to the corresponding values in the caption.

Dimensionality reduction

The structure of the data can be examined with dimensionality reduction. When data is reduced with PCA the resulting model can transform arbitrary data into the same PCA space. When other data is transformed into the same space the distributions of the upper principal components can be plotted and a visual comparison of the data is possible.

Another opportunity is to search for inherited concepts which are discovered by PCA to find new meaning in data and see how much influence single concepts have in the dataset.

Special attention has to be brought to the fact that the BFM parameter vectors are already reduced by PCA. The BFM assumes a distribution of faces based on the 200 faces it has been build upon. The PCA from parameter vectors of predicted images would be equal to the data itself if the distribution of the BFM is the same as the distribution estimated by our predictions. The principal components can only be equal, resulting in the same PCA, if these conditions are met.

- Both distributions posses the same covariance matrix

OR

- The BFM distribution and the distribution of images are equal. Or both distributions are representative samples of the population.

AND

- The BPRN does ideal predictions

If the same data is reduced ones more with PCA the resulting transformation is the same as the data itself. This can be easily comprehended because the data is already optimized for greatest variance on every dimension and a further optimization will yield the same result. When a consecutive PCA transformation changes the data the distributions are different. But the equality of both data distributions is not sufficiently proven if a consecutive PCA yields the same result. As previously shown with Anscomb's quartett different datasets can have very similar covariance matrices and will have the same PCA. With enough samples and few outliers this event should however be improbable.

Goodness of fit

The comparison of supplied data with the Kolmogorov-Smirnov test for the generative models and sampled data from the generative models will show if the generative models fitted the data well. One caveat is though that the data can only be compared parameter-wise. A bad fit can be identified if the parameter-wise comparison already fails the test.

On the other hand if the test is successful the information is not sufficient to concluded that the multidimensional distributions are equal.

To take the ks statistic into perspective, the worst score possible is a 1.0. This can only happen if either the first argument (distribution or data samples) already hit 1.0 with E_N and the second argument is still at 0 or the other way around. This would indicate that the distributions do not produce the same values at all.

Face recognition

The most essential part in this thesis is to distinguish good sampling methods from bad ones. A good prediction for an input image will create a BFM instance which looks similar to the face in the image. A similar looking BFM instance will be valued with a high similarity score when compared to the original input image by face recognition.

Although the idea of measuring the quality of a reconstructed face with face recognition appears intuitively correct, it has to be shown that face recognition is able to discriminate good from bad reconstructions. In Figure 4.6 three fittings of faces can be seen with a low similarity score on the left, a mediocre score in the middle and with a high score on the right. The right reconstruction indisputably is a good match when compared to the original image. Contrary to that the left fit does not resemble the original image.

(a) Bad fit: score 0.55 (b) Moderate fit: score 0.79 (c) Good fit: score 0.96

Figure 4.6: Examples for very high and very low scores

In Figure 4.7 the distribution of similarity scores for images created of inferenced parameter vectors from one regression network (later called BPRN[1]) are shown. It is separated into scores for predictions of the same image compared to the original (blue) and scores for every other image compared to every predicted face (orange). The distribution for different faces has a mean of 0.685 and is normal distributed, which is expected for random faces scored with each other. The blue distribution conversely does not resemble a normal distribution and is skewed to the left. This skewness may result

[1]Basel Face Model Parameter Regression Network

from images which are challenging for the regression network, be it an extreme posture or difficult lighting conditions.

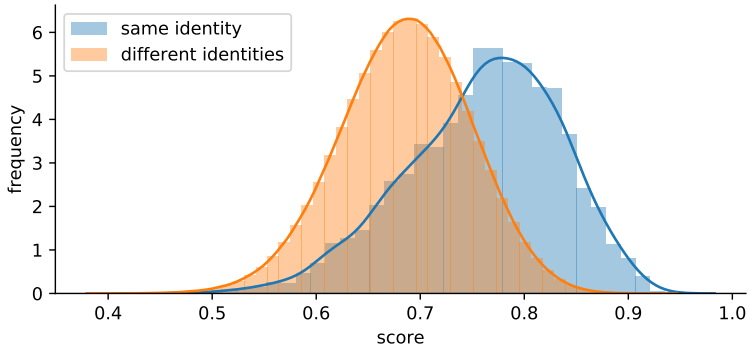


Figure 4.7: Scores for all pictures generated compared to the original input image and compared to all other original images

With these observations it can be concluded that face recognition is able to discriminate good parameter vectors from bad parameter vectors inferred by a regression network.

One notable fact is that the encodings for face recognition and the BFM can be distinguished similarly with distance. Both encodings have in common that similar vectors point to similar facial identities (facial similarity $\propto$ vector distance). For face recognition it is also appropriate that every facial identity has a unique encoding.

5 Approach

The project was approached in a bottom up fashion. In Figure 5.1 the conceptual outline of the project is illustrated. First a suitable picture dataset was acquired then a fitting algorithm was ascertained. Various generative models where examined and fewer were tested for the project. The GMM and a GAN were the best fit for the task. A new variant of GAN was then created with two continuous input variables, age and sex. The generative models are based on a different sets of good BFM instances. Several datasets for the machine learning task were then created from a variety of different randomly sampled parameter vectors and from the generative models. The regression network has been developed and trained with the created datasets. To evaluate the quality of the training data the performance of the regression nets was measured by comparing images of the predicted models with the original input image via face recognition.

5.1 Image datasets

The main source of pictures is the wiki/imdb dataset [34]. With 62.328 pictures the wiki part is considerably smaller than the imdb part which accounts for 460.723 images. For the evaluation of the different regression nets only a part from the wiki set is used. This has been done because the imdb dataset has noisy labels for age, sex and does not always show the same person for the same id or not even a person at all but an animated character. If the scraped original image depicts two persons from which one is the person under which the resulting image is filed under, the other person may appear in the cut out picture. The overall image quality is better for the wiki dataset where almost all pictures are from persons facing the camera directly. Still not all pictures were kept from the wiki dataset, due to challenges encountered while using the later described fitting algorithm. Facial features like beards or unusual skin conditions and also obstructions in the form of hair, glasses or a hand did result in a worse fit and lead to significant errors. An even more challenging problem is the ambivalent interpretation of the interaction between texture and illumination. A skin tone may be interpreted as dark when an image is dimly lit but the same skin tone can be interpreted as light in

brighter conditions. To accommodate for these challenges the images which have been chosen to be fitted were sorted constraining the set of images to good lit, no obstructions, no prominent beards and no intense makeup.

5.2 Fitting images

The initial fitting was done with the proposed probabilistic fitting algorithm from Schönborn et al. in [35] (further mentioned as p-fit). With this algorithm fitting a 3D model to an image is done in an iterative process, changing shape, texture and lighting and the expression of the model to best resemble the given image. The fittings were done with pictures from the Wikipedia face dataset [34] a preliminary step was to find facial landmarks this was handled with Dlib [27] and the shape predictor model from [36]. A total number of 4279 pictures were fitted, these models were sorted by visual evaluation and separated by visual examination into 3 categories good (1884), moderate (457) and bad fittings (1938). This discrimination method is flawed but does reasonably sort out obvious bad fittings.

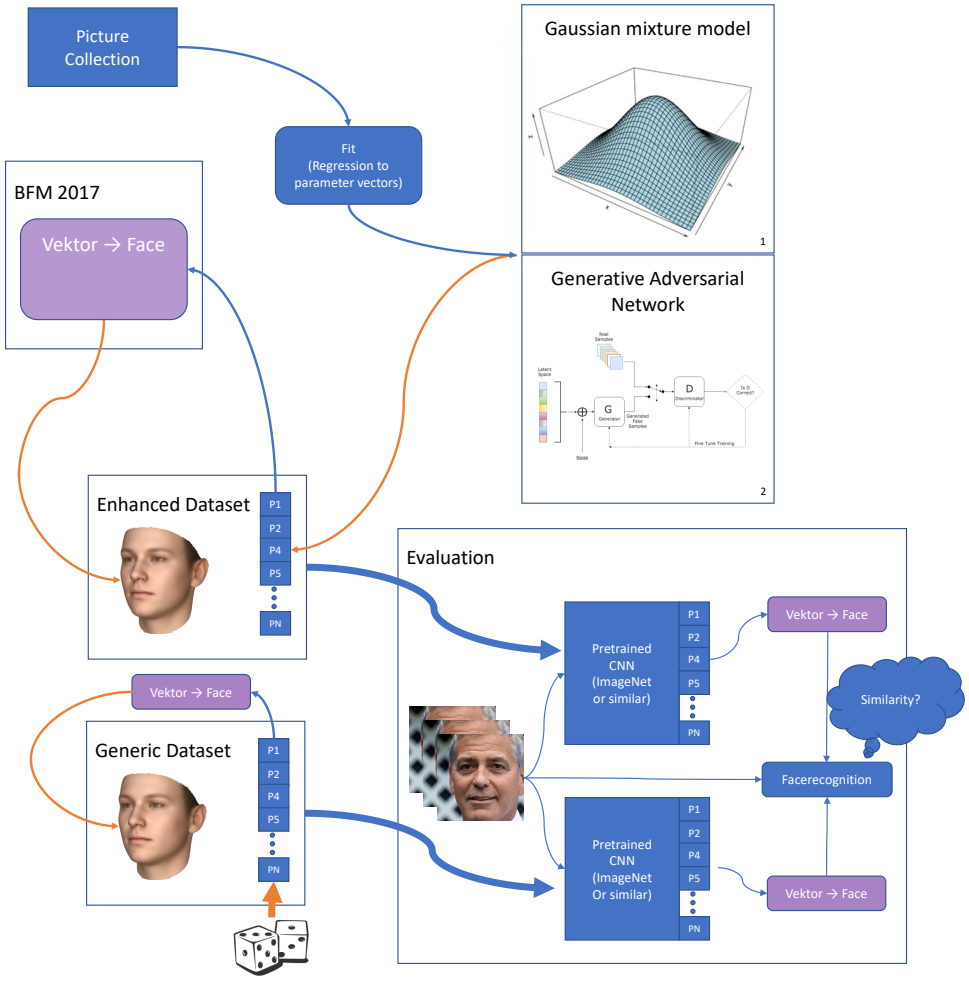
The later outlined regression network also inferences BFM instances from images and can also be considered a fitting algorithm in its own right. During the evaluating of the results from the regression network a dataset performed unexpectedly well, even better than the proposed fitting algorithm. Predicted parameter vectors from this model and also an aggregate of the best parameter vectors from a large set of models were used to train the generative models. The images used are a subset of 10000 from the wikipedia dataset. The inferred parameter vectors are further called “wiki ensemble dataset”.

5.3 Creating images

Since the proposed method of providing training data was in the form of images, which are created directly from the as label provided parameter vectors, images had to be created from these parameter vectors. This has been done with the parametric-face-image-generator[1] proposed in [3, 37] which uses the BFM to create images. The program was slightly altered to accept parameter vectors in form of json files and created corresponding pictures. For the illumination the illumination prior proposed by Egger et al. [38] is used.

[1] https://github.com/unibas-gravis/parametric-face-image-generator

Backgrounds were inserted from the MIT Places 205 test dataset [39], it consists of 40.261 images. This comparably small dataset has been chosen due insufficient memory but should still be satisfactory to create small datasets with up to 100.000 images. If a larger amount of images is to be generated the full dataset of 2.5M images should be used.



[1] https://upload.wikimedia.org/wikipedia/commons/6/6f/Screenshot_at_Juni_16_11-19-45.png
[2] https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

Figure 5.1: conceptional outline

5.4 Generating unstructured random Data

A baseline for comparing the different generative models is created with purely random generated datasets. These datasets are created with several different random distributions. They should perform poorly compared to the datasets created by the generative models. The generative models should have greater knowledge about the latent space and should create valid faces with higher probability for vectors with larger distance from the center.

This method of sampling can be compared to shooting arrows in the dark, with enough arrows there should be a good amount of hits but most arrows should fail to hit targets. In figure 5.2a are all means, standard deviations and also the distribution of the first parameter shown.

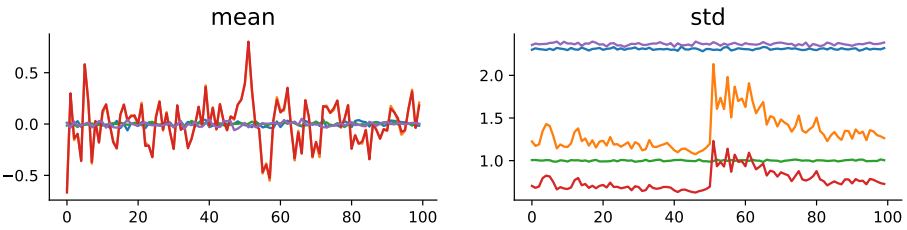
Several different sampling methods were examined the best performing are presented here. In Table 5.1 the methods and parameters are shown. Two methods sample with a zero-mean uniform and normal distribution, two use the learned means and standard deviations from p-fit and distribute uniform and normal. The last method consists of parameters 80% normal distributed and 20% uniform distributed. A visual support for the nature of the distributions is shown in Figure 5.2b.

mode	distribution	attributes
naive	uniform	$\mathcal{U}(-x, x)\ , x \in [3.5, 5]$
	normal	$\mathcal{N}(\mu, \sigma),\ \sigma \in [1-2], \mu = 0$
composite	normal + uniform	80% $\mathcal{N}(\mu = 0,\ \sigma = 1)+$ 20% $\mathcal{U}(-6, 6)$, columns shuffled
	normal + uniform (beta)	$\mathcal{N}(\mu, \sigma) \subset beta(\alpha(3), \beta(15)) \times \mathcal{U}(-6, 6)$
	normal with $\vec{\sigma}$	$\sigma = 0.7, \mu = 0$ $\mathcal{N}(\mu, \vec{\sigma}),\ \sigma_n \in [0.5, 2]$[1]
from fitted data	uniform	$(-3 \cdot \overrightarrow{\sigma}_f,\ 3 \cdot \overrightarrow{\sigma}_f)$[2]
	normal	$\mu = \overrightarrow{\mu}_f\ , \sigma = \overrightarrow{\sigma}_f$[2]

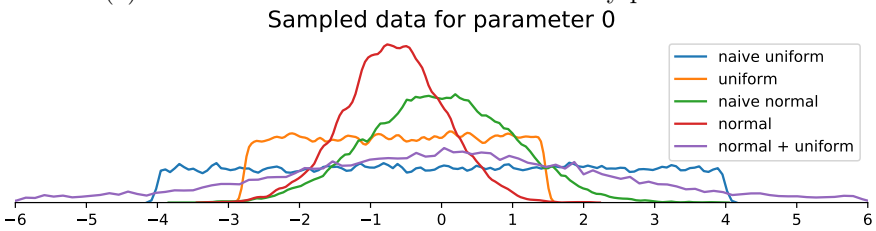
[1] in this method σ is varied inside the dataset

[2] $\overrightarrow{\sigma}_f$ and $\overrightarrow{\mu}_f$ represent the measured means and standard deviations for every parameter on basis of the p-fit

Table 5.1: Parameters for the random distributions used



(a) Means and standard deviations for every parameter



(b) Sampled data for parameter 0

5.5 Models

5.5.1 BFM Parameter Regression Network

For the neural net which is to be trained with the generated learning data the WideResNet50 [40] has been used as stem. The WideResNet50 is used for image classification and is set up with pretrained weights and biases learned on the ImageNet [41] dataset.

The net is extended with 3 fully-connected layers each 200 neurons with leaky ReLU activation and a last layer with 100 neurons corresponding to the parameter vector with tanh activation.

The decision has been made to leave the stem untrainable and only train the new last 3 layers. An experiment if training the stem for a small training dataset is sensible was made.

The net was trained with 100000 images of 10000 BFM instances with varying conditions for pose expression illumination and background. Each net was trained 10000 epochs and then evaluated. Training data and input images are previously exposed to face detection done with MobileNetV2 [42] The face detection is used to crop out the face, each image is then scaled to 224 x 224 pixels.

The WideResNet50 (see Figure 5.3) is based on residual blocks which double the filters after 1, 4, 10 and 13 blocks. Every block consists of 3 components which itself consist of a convolutional layer a batch normalization layer and a ReLu activation. The block which doubles the filters has an extra component on the residual loop.

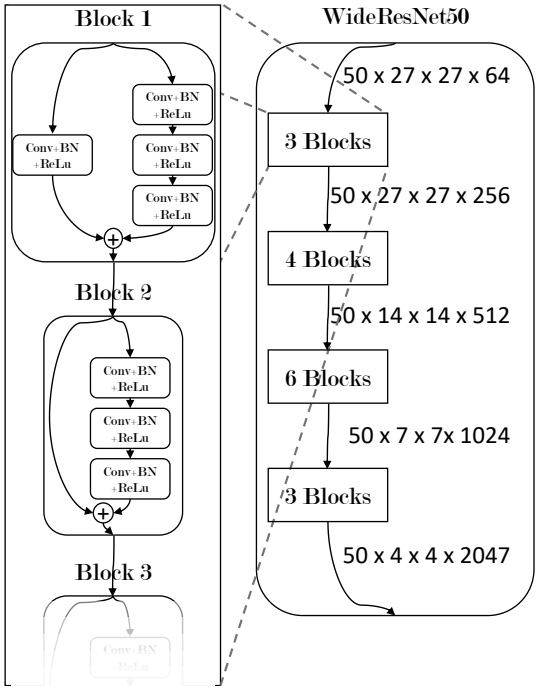


Figure 5.3: WideReseNet50 architecture. Numbers: batch size × 2d feature map (x × y) × filters

5.5.2 Generative Adversarial Network

Due to the constrains imposed by the BFM, faces with ages under 18 and over 80 will not be considered. A fit of a younger or older face may look realistic but a considerable amount tends to move into a caricatural direction. The latent vector Z is chosen with 20 parameters and additional 2 parameters for age and sex which are present in Z and X_{fake}. Age and sex labels are provided by the Wikipedia dataset and are included in X_{real} and X_{fake} together with the parameter vector for a face. The conceptual architecture of the GAN is shown in Figure 5.4. The layer composition for D and G is shown in Table 5.2 and 5.3. The used data is comprised of the best predictions from all computed BPRN models. The loss functions for generator and discriminator are formulated as proposed by Goodfellow [43] with optimized loss for D to avoid vanishing gradients early on in training. Creating a GAN with few learning data requires a decision to either overfit the discriminator or more heavily regularize the model. Either way has

	input	operation	activation	
	200×22 ($2 \rightarrow$ age/sex)	fully connected	leaky ReLU	
⎛	200×2000	fully connected	leaky ReLU	⎞
⎜	200×2000	dropout 50%		⎟ CVGAN1
⎜	200×1000	fully connected	leaky ReLU	⎟
⎝	200×2000	dropout 50%		⎠
(	200×500	fully connected	leaky ReLU	) CVGAN2
	200×500	fully connected	leaky ReLU	
(	200×500	dropout 50%		) CVGAN1
	200×100	fully connected	tanh	
	$200 \times 100 + 200 \times 2$ (age/sex)	concat		

Table 5.2: GAN: Generator layers

	input	operation	activation	
	200×22 ($2 \rightarrow$ age/sex)	fully connected	leaky ReLU	
⎛	200×2000	fully connected	leaky ReLU	⎞
⎜	200×2000	dropout 50%		⎟
⎜	200×1000	fully connected	leaky ReLU	⎟ CVGAN1
⎝	200×1000	dropout 50%		⎠
⎛	200×500	fully connected	leaky ReLU	⎞
⎝	200×500	dropout 50%		⎠ CVGAN2
	200×500	fully connected	leaky ReLU	
	200×500	dropout 50%		
	200×100	fully connected		

Table 5.3: GAN: Discriminator layers

the same negative effect that the discriminator does not generalize well. Dropout has been used for the discriminator to regularize but not for the generator.

$$\mathcal{L}_D = -\log(D(X)) - \log(1 - D(G(Z)) \tag{5.1a}$$

$$\mathcal{L}_G = -\log(sigmoid(D(G(Z)))) \tag{5.1b}$$

Because the network has been trained to learn a representation of an identity which is invariant to age and sex, this representation should be distinguished from the usual concept of the identity of a face. The sex of a person is fixed and therefore should be a conditional feature. Optimally the identity of a face is preserved while the age could be varied.

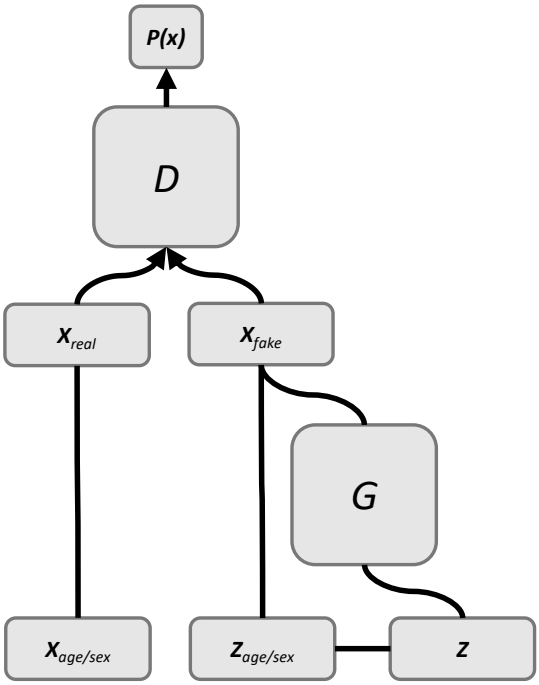


Figure 5.4: GAN architecture

Two different models of the GAN have been modeled and samples of both have been used to train the BPRN. The models do differ in the use of noise and dropout. The first model (M_1) with three layers of 2000, 1000 and 500 neurons has dropout for the discriminator and the generator and does not use noise. The second model (M_2) does have only two layer with 500 neurons each and input vectors were changed with normal distributed noise $\mathcal{N}(\mu = 0, \sigma = .1)$.

5.5.3 Gaussian Mixture Model

As stated the GMM assumes a multinomial Gaussian distribution for the provided data. This constraint seems apparently to be met by the parameters regressed with the probabilistic fit algorithm (see Figure 5.5). It can be assumed to the best of knowledge that regressed BFM parameter vectors follow a normal distribution. The application of a GMM therefore is appropriate.

The data used for the creation of this model is the collection of the highest scoring parameter vectors from 13 BPRNa which predicted 11000 pictures from the wikipedia dataset.

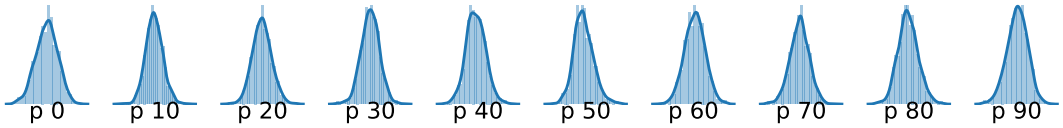


Figure 5.5: Distributions for 10 parameters from the p-fit dataset

Components	AIC	BIC
1	3.13e+06	3.17e+06
5	3.02e+06	3.20e+06
25	3.07e+06	3.99e+06
100	1.85e+06	5.54e+06
400	-1.26e+06	1.34e+07
1000	2.92e+06	3.98e+07
4000	3.18e+07	1.79e+08

Table 5.4: Gaussian Mixture Models with corresponding AIC and BIC scores

The number of components for the GMM is the most significant parameter, it does determine if the model will over- or underfit the data. The AIC and BIC scores suggest that either 1 (for BIC) or 400 (for AIC) components are suitable for the model (see Table 5.4). Another method to determine the goodness of the model is to compute the PCA for the original data and use the transformation matrix to plot samples from the created GMMs. In Figure 5.6 the original data and samples from all proposed component sizes are plotted into the first two dimensions of the PCA computed from the original data. The shape of the original data has one center and spreads out in a u shaped pattern. For component sizes 1 to 5 this can not be observed, but beginning with 25 components the shape starts to resemble the original shape of the data. The changes to the shape of the data distribution do decrease with higher component size. On visual inspection alone a GMM with 100 components should sample decent data for our purpose.

All datasets created with the explained methods are used to train BPRNs. After training the models will predict the same images which have been considered a good fit by p-fit. The predictions, in form of parameter vectors, are computed to pictures and are evaluated with face recognition by comparing each prediction with the original. For face recognition VGG-Face CNN [44] is used.

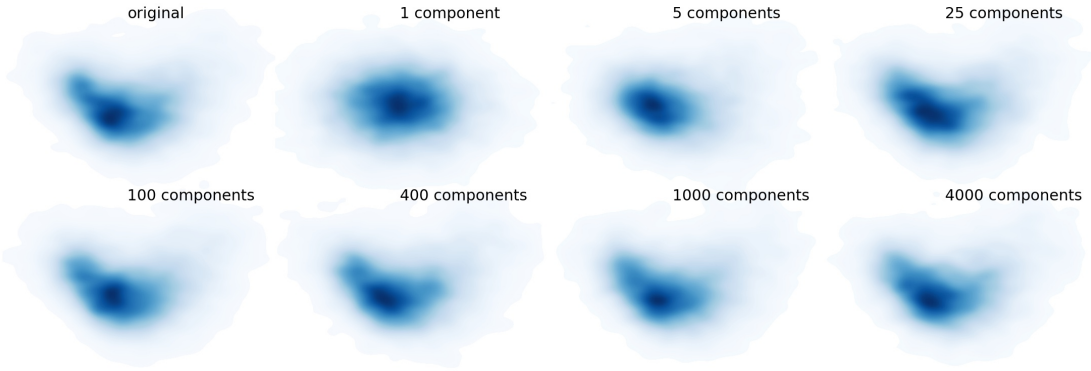


Figure 5.6: Comparison of GMMs with changing component size. The original data is plotted to the first two dimensions of it's PCA. From every GMM samples are transformed to the same Space and also plotted to the same two dimensions.

6 Evaluation

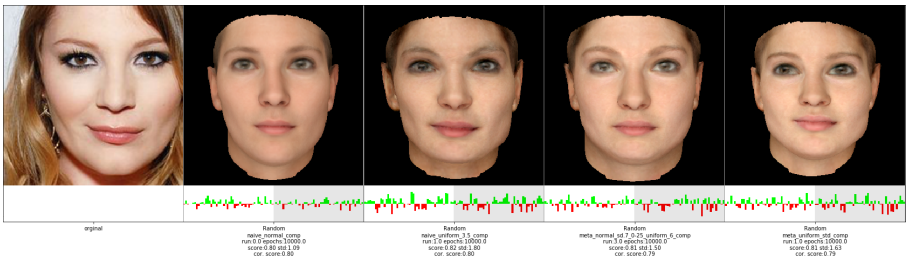
6.1 BFM Parameter Regression Network

The BPRN made satisfactory predictions with almost all datasets. But it lacked versatility in some areas. The predicted parameter vectors are not decoupled from several side effects. These effects are: illumination, facial expression, pose and occlusion. Examples for all effects are showcased in Figure 6.1. This predicament does exist due to the constrained input of only one image, without multiple inputs of the same identity under different conditions the extraction of the identity is arduous.

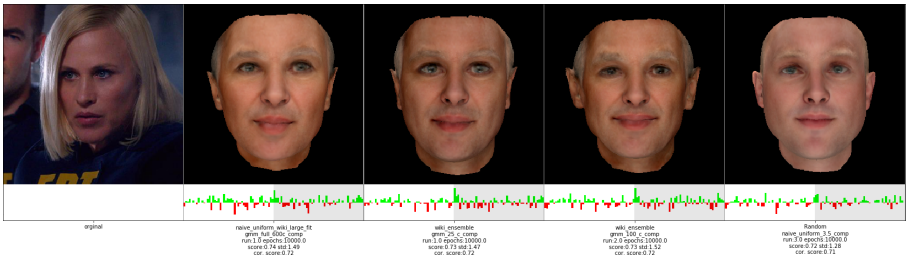
The decision to use an existing net as the stem for the BPRN makes the BPRN less flexible. The new net can fall back on the experience of the network beneath, but the upper net can only build on the features the WideResNet50 asserted as valuable for the task it was trained on. Deep neural nets can learn high abstractions, in [45] it is shown that a CNN that has been trained for classification on the ImageNet dataset did develop layers with high activations for faces not only human faces but also from animals. While a learned affinity to faces helps the developed BPRN a large part of learned abstractions are of no value.

An experiments has been realized to test the viability of training the stem of the network. In Figure 6.2 the effects of training the steam and only training the last appended layers is shown over 2000 to 100000 epochs trained. The numbers for this Figure are in Table 6.1. Both methods increase standard deviation logarithmically per instance with longer training. When training the stem the standard deviation per instance is significantly lower and may not reach such high values as training with the stem fixed.

The optimal corrected score and raw score for training with fixed stem peaks around 6k - 10k epochs and then declines. For training including the stem the corrected score and raw score peak later and may also increase with longer training than 100k epochs.

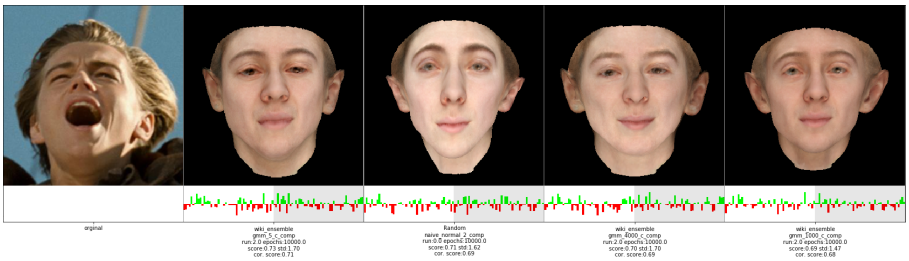


(a) Expression: expression is leaking into model. here: left corner of the mouth pulled up

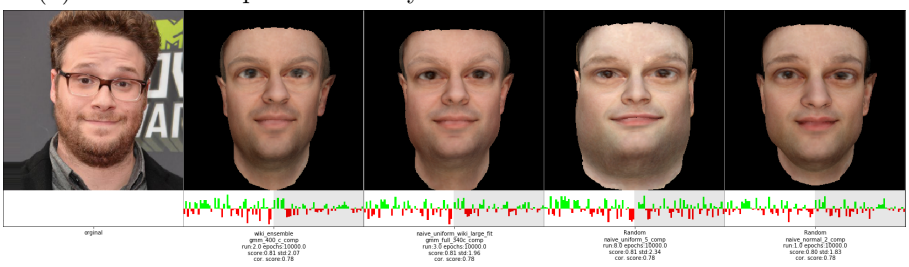


(b) Illumination: unusual illumination leads to wrong skin tone

(c) Occlusion: occlusions like hands are incorporated and result in distortion, here bigger chin and neck



(d) Pose: intense poses are badly translated and result in a skewed face



(e) Glasses: glasses are generally interpreted as large eyes

Figure 6.1: Systemically wrong prediction by the BPRN

Train stem	Epoch	corrected score	corrected rank	score	rank
False	2000	.763	63.67	.764	73.76
	4000	.770	57.84	.774	66.04
	6000	.780	47.27	.789	51.26
	8000	.773	54.76	.784	55.49
	10000	.778	49.32	**.790**	**49.13**
	20000	.770	57.60	.787	52.81
	30000	.769	59.04	.786	54.15
	100000	.770	57.66	.788	51.55
True	2000	.755	71.14	.755	80.25
	4000	.768	59.88	.768	70.67
	6000	.774	53.79	.775	64.28
	8000	.780	47.60	.782	58.12
	10000	.744	73.22	.745	80.19
	20000	.782	45.24	.785	54.89
	30000	**.784**	**42.73**	.787	52.17
	40000	.782	45.25	.786	54.18
	100000	.783	44.68	.786	53.24

Table 6.1: Corrected scores and ranks for "normal with $\vec{\sigma}$" with only the last 3 layers trained and with the full net including the stem (WideResNet50).

The training including the stem has large overfitting which is already apparent after 2k epochs. This does not happen with a fixed stem, the loss for test and training data stays similar even after 100k epochs.

Because we are using synthetic data it can be argued that overfitting synthetic samples may not negatively impact generalization for real input images. But it has to be taken into account that artificial faces inherit the same abstract concepts as real faces do. So overfitting in this instance is still unfavorable for good predictions.

The BPRN could be rebuild from scratch without a pretrained stem. This would lead to longer training times but may build better abstractions in deeper layers.

Training the BPRN without pretrained parameters is not a viable alternative because training takes even longer than when training with pretrained stem. One experiment has been done and after 10k epochs of training the predictions of the BPRN only scored with a mean of 0.7 which is the same for random faces matched against one another. The per instance standard deviation was also only at 0.5 which results in only in very average faces.

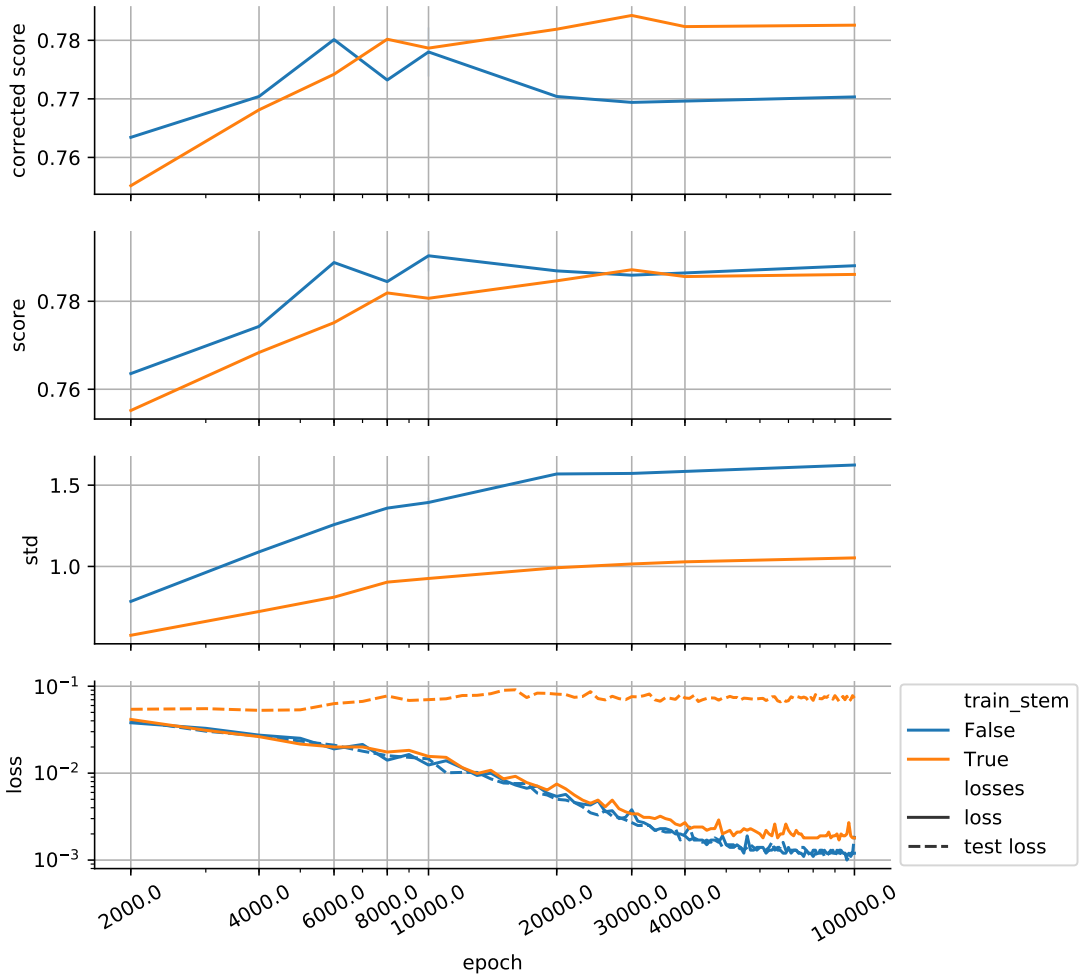


Figure 6.2: The mean scores and corrected scores of the neural net based on the dataset "normal with $\vec{\sigma}$" plotted for epochs learned, with only training of the last appended layers or the full net. The bottom row shows the loss and test loss for the 100.000 epoch training with and without full training.

The BFM itself is not fully decoupled from all non identity related influences, as shown under 3.1.1. A decoupling of illumination, pose and expression in the BPRN would be partly possible when the labels are extended with information of expression, pose and illumination. These informations are present in the image generation process and can be extracted. The practicality of the label extension is compromised by the incorporated illumination and expression in the BFM for shape and texture. This may not make it possible to satisfiable decouple illumination and expression even with extended labels.

6.2 Generative Models

The generative models created in this project were a set of Gaussian Mixture Models and two Generative adversarial Networks with similar architecture. These models were used to sample training data for the BPRN. Both methods are performed with a selection of BFM instances with a high similarity score for the image they were predicted for. We first look at what the models distinguishes and then how well both sample the provided distribution.

6.2.1 Generative Adversarial Network

The GAN has been developed to sample parametric faces. The faces can be adjusted for age and sex. In Figure 6.3 faces with the same parameters for the input Z can be seen with different values for age and sex. A transition from young to old (horizontal) and from female to male (vertical) can be observed.

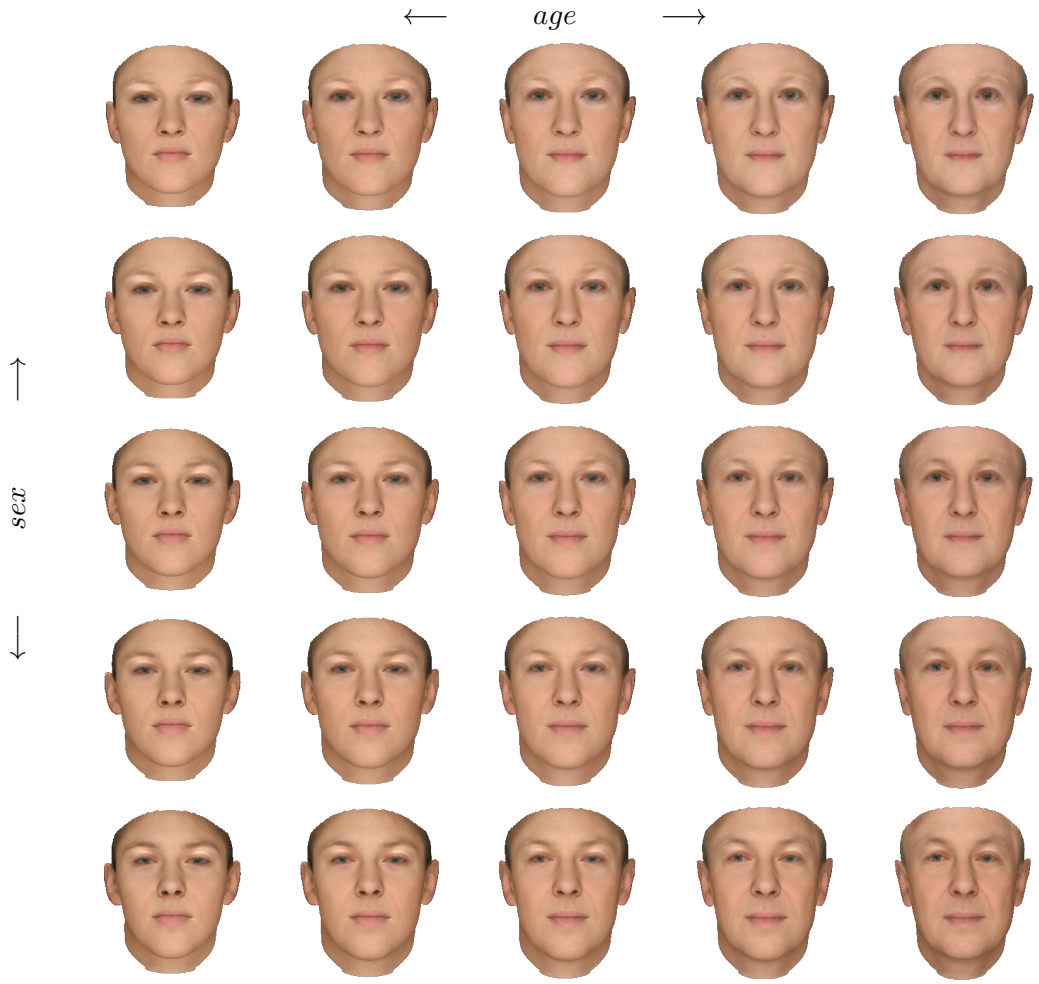


Figure 6.3: Latent space of the proposed GAN iterated for learned age and gender from 18 years to 80 and from female to male. It can be seen that shape and texture are in line with changes in age and sex, for example wrinkles and gray hair are present for older ages

6.2.2 Gaussian Mixture Model

The GMM was modeled for different component sizes and used to sample BFM instances for the training of the BPRN. The component size for the GMM influences the goodness of fit distinctively with more components. In Table 6.2 and Figure 6.4 measures for different component sizes are shown. Similarity scores corrected and raw do not significantly differ between models. A slight increase in standard deviation per instance and a potentially linked increase in raw scores for more components is visible. The concept of corrected scores will be explained later. The scores for AIC and BIC indicated either 400 or one component to be the best decision.

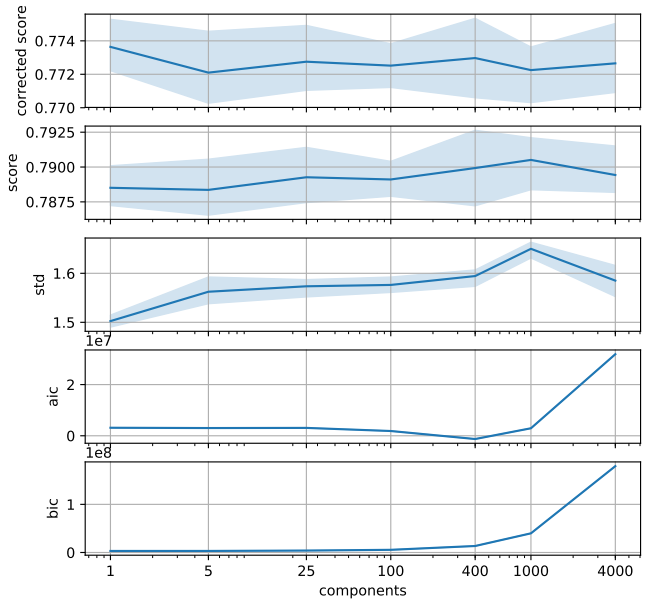


Figure 6.4: GMMs compared by component size, with 95% confidence interval for scores and standard deviation per instance.

components	std	corrected score	corrected rank	score	rank	aic (10^6)	bic (10^6)
1	1.5	0.7781	54.28	0.7885	52.45	3.10	3.17
5	1.56	0.7769	55.63	0.7884	52.58	3.02	3.20
25	1.57	0.7777	54.66	0.7893	51.53	3.07	3.99
100	1.58	0.7775	54.85	0.7891	51.63	1.86	5.55
400	1.59	0.7780	54.52	0.7899	51.17	-1.26	13.48
1000	1.65	0.7777	54.80	0.7905	50.35	2.93	39.81
4000	1.59	0.7776	54.70	0.7894	51.38	31.87	179.39

Table 6.2: GMMs compared by component size

6.2.3 Comparing Generative Models

Complexity

The two models each have a set of variables to compute samples, the number of variables used for the model reflect the complexity of the model. This has effects on expressiveness, memory usage and computation time. The GMMs vary widely in the size of variables, mainly because the size of components is linearly tied to the complexity. A covariance matrix is computed for every component with 100 rows and columns. The GMM with 4000 components is 4000 times larger in size of the used variables than the GMM with 1 component.

The GANs use weights and bias, for every fully connected layer, this results in $2 \times (L_{n,size} \times L_{n+1,size})$ variables per every consecutive fully connective layer.

The larger GAN is on par with the GMM 400 and the smaller GAN is between GMM 25 and GMM 100.

Goodness of fit

To evaluate the generative models we initially compare the sampled data with the provided data. The provided data chosen from the best ranking instances that have been generated by BPRNs that have been trained with structureless random datasets. To compare the similarity of two datasets the two-sample Kolmogorov-Smirnov test is a good measure. A drawback is that only one dimensional distributions can be measured. This means that only one parameter at a time can be compared, but this still gives some insight about the quality of the fitted model. and the ks statistic describes the disparity between the two datasets. We measure the ks statistic for every parameter at a time and then compute the sum.

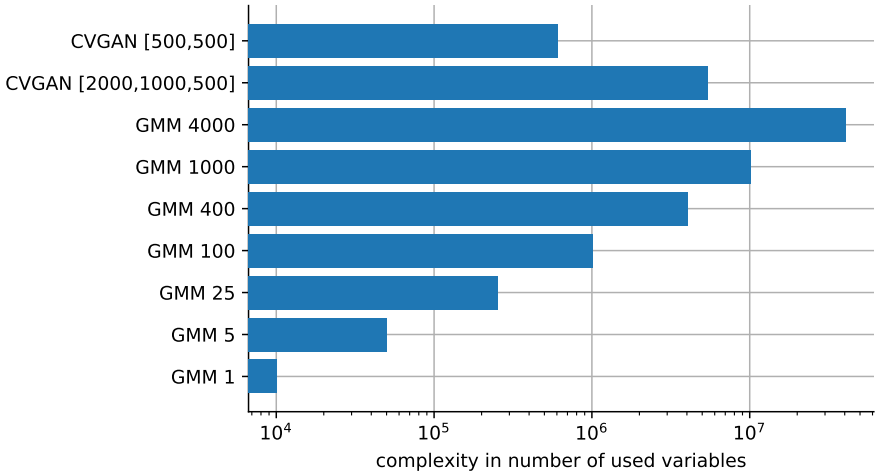


Figure 6.5: Complexity of the generative models measured in used variables

$$KS = \sum_{p \in P} \sup_x |g(x) - f(x)|$$

The computed results can be seen in Table 6.3. When comparing the GMM samples the trend stands out that with higher component size the ks statistic decreases. This is plausible as a higher component size results in a better fit to the supplied data. The mean p-values for all GMMs are high and imply that we can't reject the hypothesis that both datasets come from the same distribution. The GANs on the other hand do produce significantly higher values for the KS statistic and also have very low p-values. The first model with 3 layers does particularly bad with a 10 times higher KS statistic as the GMMs and a statistically significant p-value of 0.001. The second GAN with 2 layers does also worse than all GMMs and also as a statistically significant p-value. Which indicates that the null hypothesis can be rejected and we can be sure that the sampled data from both GANs does not come from the same distribution as the supplied data.

The sampled data from the GMMs model the supplied data distribution better than the sampled data from the GANs.

Model	∑ K-S	∅ p-value
GMM 1	1.617	0.308
GMM 5	1.826	0.196
GMM 25	1.505	0.308
GMM 100	1.340	0.407
GMM 400	1.417	0.343
GMM 1000	1.354	0.386
GMM 4000	1.187	0.551
CVGAN1 [2000, 1000, 500] G .5 drop	11.581	0.001
CVGAN2 [500, 500] .1 noise	6.756	0.005

Table 6.3: Generative models compared by sum of KS statistic of all parameters and corresponding p-value

6.3 Measuring data variety

Generally when creating training data the goal is to reproduce an existing distribution of data. As previously stated the event space of faces is large and sampling faces from it should result in distinctly different instances. With sampling from uniform and normal distributions the variety of the instances is ensured. By learning the structure of fitted faces and sampling from generative models parts of the events space are omitted. To measure the variety in the created models three methods are used to evaluate the created samples, clustering, dimensionality reduction and distances.

6.3.1 Clustering

If distortion degrades slowly with increasing clusters the data is well spread out, if distortion decreases fast the data inherits more clustered data points. For a baseline example the distortion of data sampled from a uniform distribution is shown in Figure 6.6 for comparison the distortions for GMM 4000 are also plotted. The change in distortion for all datasets is shown in Table 6.4. It can be seen that purely random uniform or normal distributed datasets decrease to about 9.36 at 200 clusters, this also holds true for mixed random normal and uniform approaches which are based on means and standard deviations of fitted data. Compositions of several random distributions tends to decrease the distortion more (see the last 3 entries in Table 6.4 with 3 or more distributions). The generative models however produce samples which have immensely decreased distortion. The GMMs approximate the same values for distortion as the provided data but the GAN's seem to create much more clustered samples than the

data provided does. The Gaussian Mixture models do decrease distortion with higher component size due to a more tight fit. The variety of samples should therefor intuitively decrease with more complexity of the model. Another notable observation is that composite random datasets partly also decrease distortion. This can be explained by the larger variation in standard deviation per instance and thus following cluster creation. If all instances have the same standard deviation they are all spread out evenly, when this is not the case clusters are forming for instances with lower standard deviation.

Source	type	2	10	25	100	200
wiki ensemble fit	data	9.59	9.14	8.93	8.63	8.44
	CVGAN1 2000x1000x500 dropout	**9.23**	**8.03**	**7.52**	**6.81**	**6.46**
	CVGAN2 500x500 noise	9.38	8.48	8.04	7.37	7.03
	GMM 1	9.71	9.37	9.19	8.92	8.76
	GMM 5	9.62	9.23	9.06	8.78	8.62
	GMM 25	9.63	9.18	8.95	8.67	8.49
	GMM 100	9.63	9.17	8.94	8.57	8.37
	GMM 400	9.59	9.14	8.92	8.57	8.35
	GMM 1000	9.58	9.13	8.90	8.57	8.36
	GMM 4000	9.57	9.11	8.91	8.57	8.33
p-fit	uniform	9.95	9.82	9.71	9.50	9.37
	normal	9.93	9.80	9.70	9.51	9.36
	uniform x 2	9.95	9.82	9.71	9.51	9.36
Random	naive normal	9.93	9.80	9.70	9.50	9.37
	naive normal 1.5	9.94	9.81	9.71	9.51	9.38
	naive normal 1.8	9.94	9.81	9.71	9.51	9.38
	naive normal 2	9.94	9.81	9.71	9.51	9.37
	naive uniform 4	9.95	9.82	9.72	9.51	9.36
	naive uniform 3.5	9.95	9.82	9.71	9.51	9.37
	naive uniform 5	9.95	9.82	9.71	9.51	9.36
Random composition	normal.8 σ 1 uniform.2 6	9.90	9.77	9.67	9.48	9.35
	normal.8 σ 2 uniform.2 6	9.93	9.80	9.70	9.50	9.37
	normal σ .7 beta 3 15 uniform 6	9.66	9.57	9.45	9.16	8.98
	meta normal σ .7 0-25 uniform 6	9.66	9.61	9.46	9.17	8.97
	meta uniform std	9.38	9.30	9.25	9.07	8.93

Table 6.4: Distortions for 2 to 200 clusters for all evaluated datasets

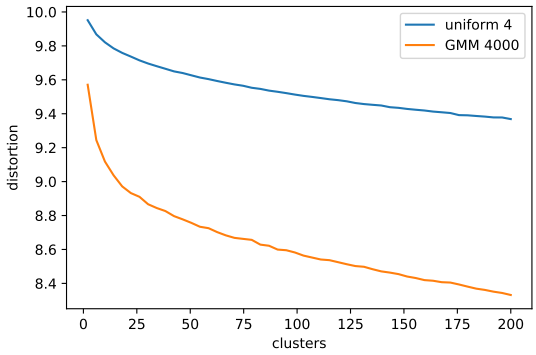


Figure 6.6: Decreasing distortion for n clusters for data sampled from a uniform distribution and the GMM with 4000 components

6.3.2 Distances

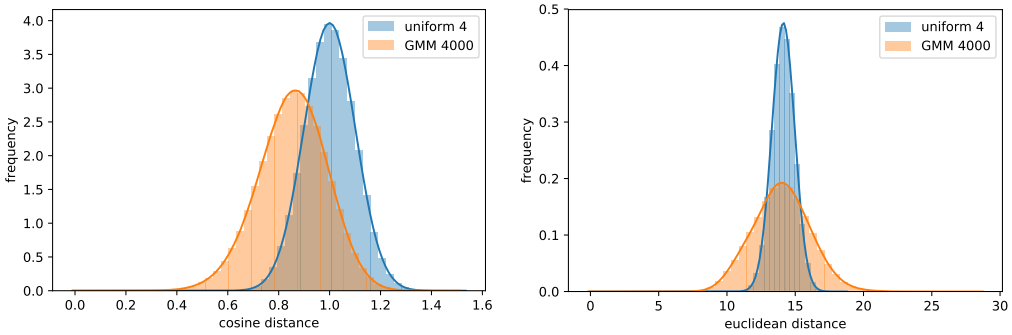
When comparing observations the measure of distance is an indicator for the distribution of the data. In Table 6.5 quartiles for cosine and euclidean distances are shown. One noteworthy observation is that datasets from generative models are all shifted to the left when compared with structureless random datasets. This indicates that samples are generally speaking more similar than orthogonal or anti similar to other samples. This observation is not implicitly a bad indicator for the quality of the data but it shows that generative models do yield more models with similar feature characteristic.

The euclidean distance is highly correlated with the per instance standard deviation of the dataset. It can be seen that difference between quartiles is about 1-3 for all datasets. This shows that the distances between samples in the dataset is largely the same.

The generative models, with the exception of CVGAN1, mirror the values for L2 and cosine distance.

Source	type	cos Q_1	cos Q_2	cos Q_3	L2 Q_1	L2 Q_2	L2 Q_1
wiki ensemble fit	data	0.76	0.85	0.95	18.28	20.21	22.11
	CVGAN1 2000x1000x500 dropout	**0.61**	**0.78**	**0.94**	16.22	18.65	21.08
	CVGAN2 500x500 noise	0.71	0.85	0.98	17.51	20.17	22.97
	GMM 1	0.75	0.85	0.95	18.88	20.16	21.56
	GMM 5	0.77	0.86	0.95	18.36	20.16	21.99
	GMM 25	0.77	0.86	0.95	18.51	20.23	22.01
	GMM 100	0.77	0.86	0.95	18.47	20.20	22.01
	GMM 400	0.76	0.86	0.95	18.19	20.04	22.03
	GMM 1000	0.76	0.86	0.95	18.05	19.99	22.10
	GMM 4000	0.76	0.86	0.95	18.08	20.15	22.18
p-fit	uniform	0.90	0.97	1.04	18.22	19.03	19.84
	normal	0.85	0.92	0.99	**10.42**	**10.97**	**11.53**
	uniform x 2	0.92	0.99	1.07	36.44	38.05	39.65
Random	naive normal	0.93	1.00	1.07	13.44	14.10	14.78
	naive normal 1.5	0.93	1.00	1.07	20.13	21.14	22.15
	naive normal 1.8	0.93	1.00	1.07	24.16	25.37	26.59
	naive normal 2	0.93	1.00	1.07	26.88	28.22	29.58
	naive uniform 4	0.93	1.00	1.07	27.41	28.55	29.69
	naive uniform 3.5	0.93	1.00	1.07	31.29	32.61	33.91
	naive uniform 5	0.93	1.00	1.07	39.09	40.72	42.35
Random composition	normal.8 σ 1 uniform.2 6	0.93	1.00	1.07	23.65	25.14	26.67
	normal.8 σ 2 uniform.2 6	0.93	1.00	1.07	31.81	33.38	34.96
	normal σ .7 beta 3 15 uniform 6	0.93	1.00	1.07	16.28	18.64	20.86
	meta normal σ .7 0-25 uniform 6	0.93	1.00	1.07	16.30	18.65	20.87
	meta uniform std	0.93	1.00	1.07	14.82	18.26	21.23

Table 6.5: Cosine and euclidean (L2 norm) distances with first, second (median) and third quartile. The lowest values are highlighted.



(a) cosine distances

(b) euclidean distances

Figure 6.7: Measured distances for random data generated with a uniform distribution $\mathcal{U}(-4, 4)$ and GMM with 4000 components

6.3.3 Dimensionality reduction

With Principal component analysis large concepts can be discovered in the first principal components. The best models predicted by all BPRNs trained with random data are evaluated. The wiki image dataset provides labels for age and sex of the persons in the images. With these labels the two visualizations are created. In Figure 6.8 the first 5 principal components are plotted with the data labeled for sex of the person in the image. On the upper diagonal a scatterplot shows a set of points from the dataset and in the bottom diagonal the mean values of all observations are plotted. The first two dimensions do inherit a very large portion of the sex concept. This revelation however was to be expected. The BFM itself also inherits the male female concept prominently in the fist component of the PCA, which is plausible because the human population can be best evenly split in male and female. The first two dimensions almost seamlessly separate both sexes, when used as a classifier with

$$sex(\boldsymbol{x}) = \begin{cases} \text{male} & if\ ||\boldsymbol{\mu}_{\male} - \boldsymbol{x}|| < ||\boldsymbol{\mu}_{\female} - \boldsymbol{x}|| \\ \text{female} & else \end{cases} \tag{6.1}$$

we receive 82.8% right classifications for sex. If all 100 dimensions are used classifications only improve to 87.6%, this further underlines that most of the sex concept is inherited in the first two dimensions.

The same visualization has been created for age groups in Figure 6.9. As opposed to sex the age concept does not separate the data sufficiently to use it as a classifier. The first dimension shows most of the concept and all following dimensions only slightly inherit the concept. It can be concluded that age and sex are both concepts which are highly influence the composition of BFM instances.

In Figure 6.10 The PCA transformation of the wiki ensemble dataset is visualized and below the PCA transformation from data sampled by the CVGAN2 is into the same space is shown. The distributions look similar while the distribution of the GAN is vertically bulged. The male/female groups sampled by the GAN show very similar positioning as the male/female groups in the original dataset. With the exception of a greater portion of female instances leaking into the space of male instances.

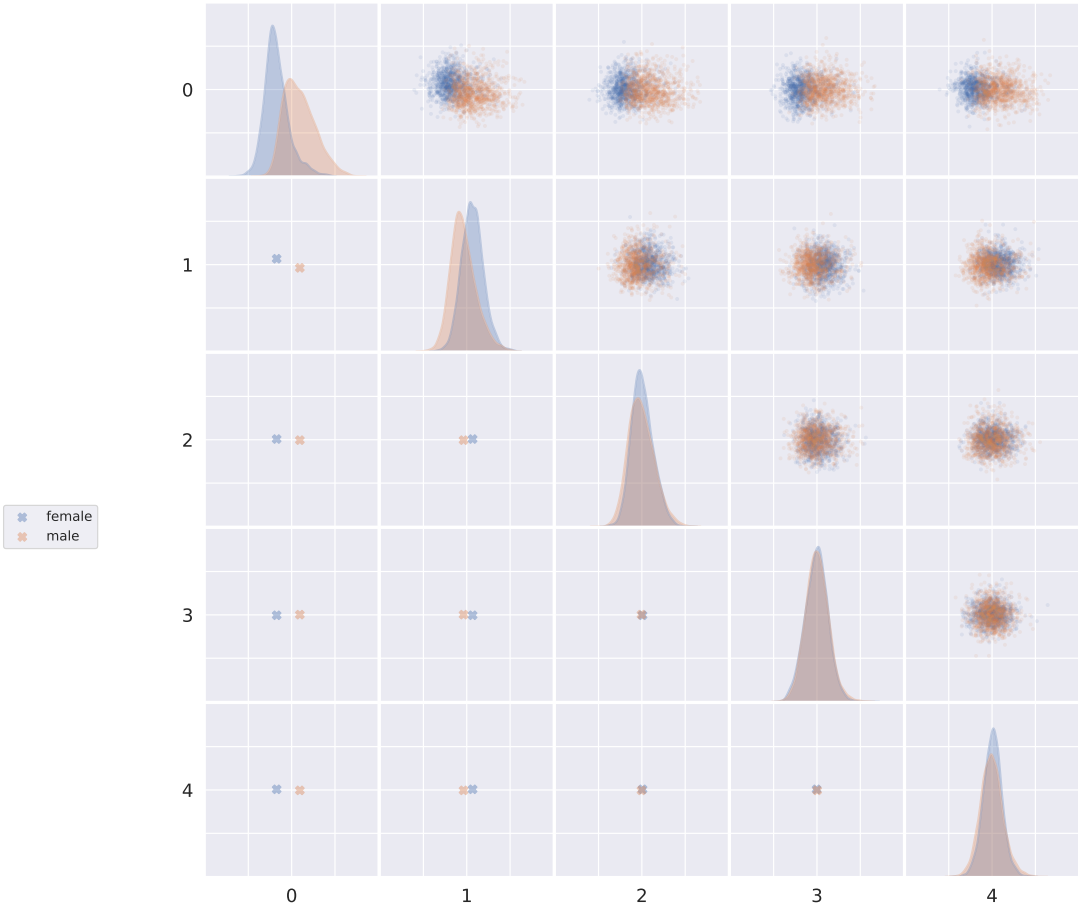


Figure 6.8: Concept discovery for the first 5 principal components of the wiki ensemble dataset, with male and female instances highlighted. Plots under the diagonal show the mean values of the categories. Plots over the diagonal show a 1000 random points from the whole dataset.

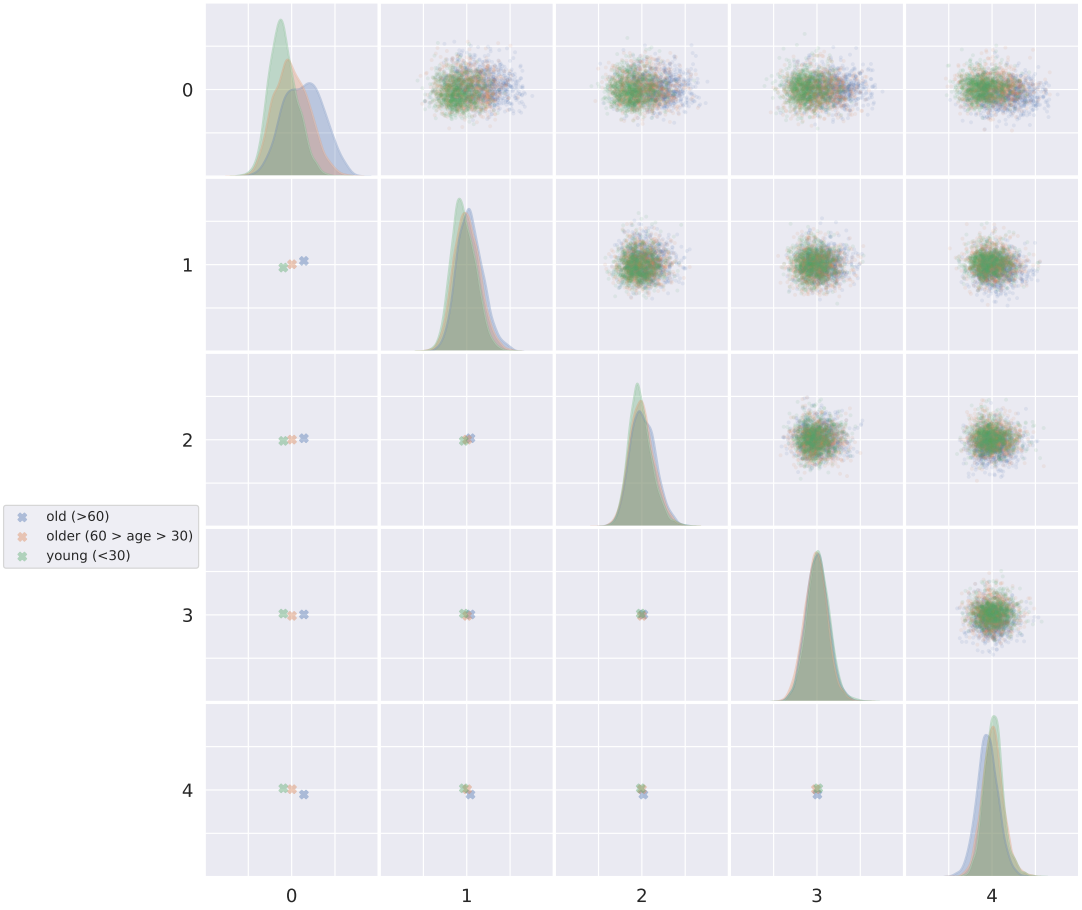
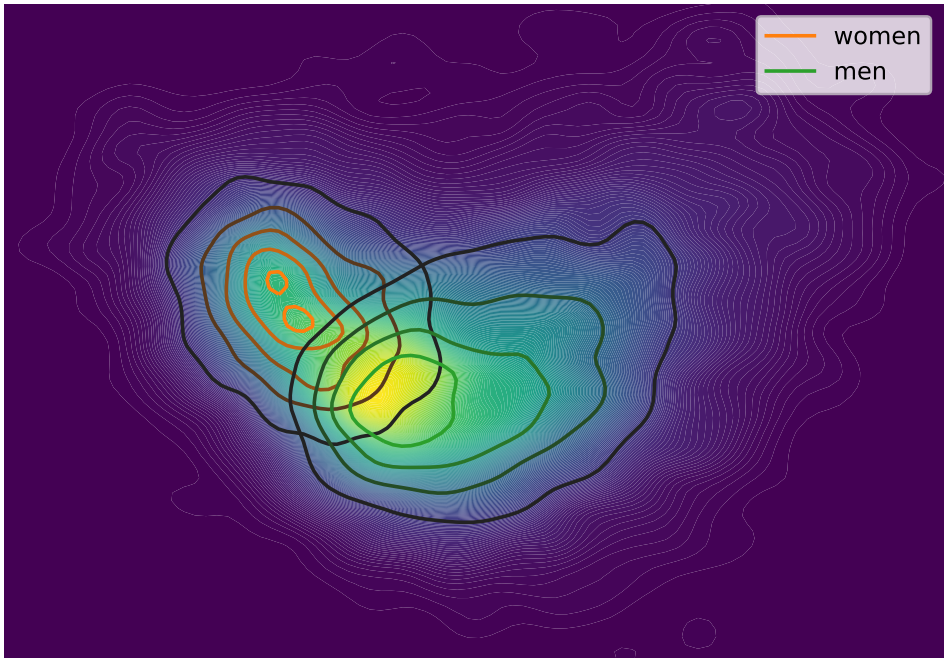
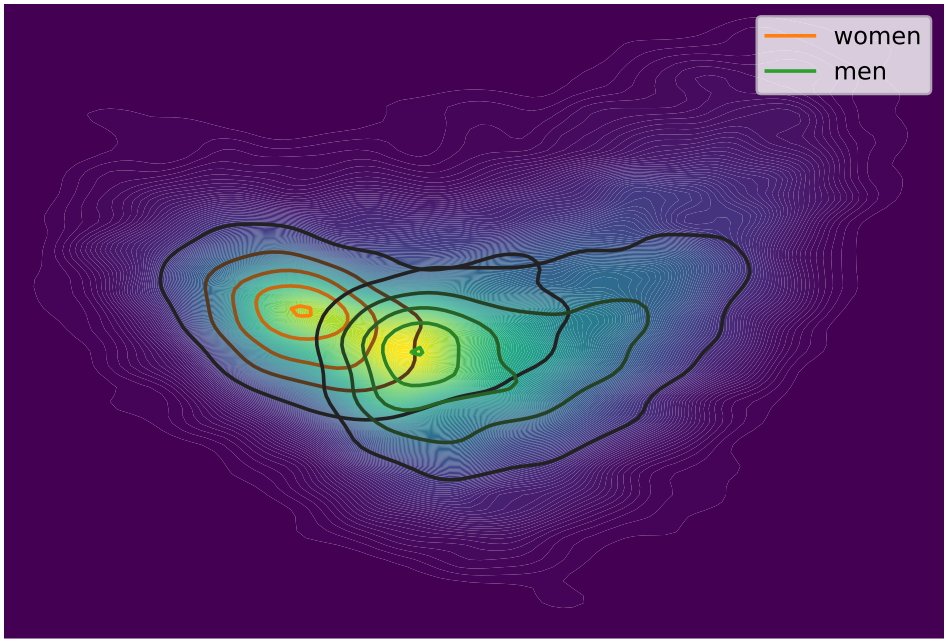


Figure 6.9: Concept discovery for the first 5 principal components of the wiki ensemble dataset, with 4 age groups highlighted. Plots under the diagonal show the mean values of the categories. Plots over the diagonal show a 1000 random points from the whole dataset.



(a) first 2 PCA components (17.8% and 9% explained variance) from the wiki ensemble dataset



(b) Data sampled from CVGAN2 transformed in the same PCA space

Figure 6.10: PCA space with distributions for female and male instances

6.4 Face recognition

6.4.1 Explanatory power of similarity scores

The higher the similarity score the better the similarity between the created output and the original image. This stays true for models which have σ up to 1, instances with greater σ have exaggerated features and are prone to look unrealistic but will be rated higher by the face recognition. This condition can be mitigated by penalizing the similarity score for $\sigma > 1$ (see equation 6.2). With corrected scores for $n = 30$ the correlation between σ and score is discontinued, as can be seen in Figure 6.11. But the corrected score should not be taken as the last truth for quality of the inferences. Values between 10 and 40 are possible candidates for n. A correlation between standard deviation and score may still exist with the optimal value.

$$f(\sigma, s, n) = \begin{cases} s & \text{if } \sigma < 1 \\ \sqrt[n]{\sigma} \cdot s & \text{else} \end{cases} \tag{6.2}$$

With the assumption that realistic faces should perform better, datasets with random faces should inherit many invalid instances and perform poorly for faces with greater standard deviation compared to datasets created with a fitting algorithm or from samples of a generative model. This assumption does not seem to hold true for randomly generated faces and the probabilistic fitting algorithm used. As can be seen in Table 6.6. Even for corrected scores with strong penalty for larger standard deviations p-fit still performs worse than uniform.
In the appendix predictions from all methods are shown. Models trained with data that has larger standard deviation are more daring to give higher estimates for parameters as models with smaller standard deviation. The uppermost image from naive uniform (in fig 5.2b) is a good example for a prediction that could be assessed better by the face recognition but apparently be worse than other predictions assessed by the human eye.

The generative models constrain the latent space and reduce the set of invalid instances. While there are may be less invalid faces the diversity suffers and large parts of the face space may be omitted. The BPRN learned to predict reasonable faces with random distributions that have a higher standard deviation, that results in faces which have overly exaggerated features (see Figure 6.12). The CNN may have profited from more exaggerated data while only learning from a comparably small dataset. This may diminish with larger datasets and a larger variety of realistic faces. Especially for face

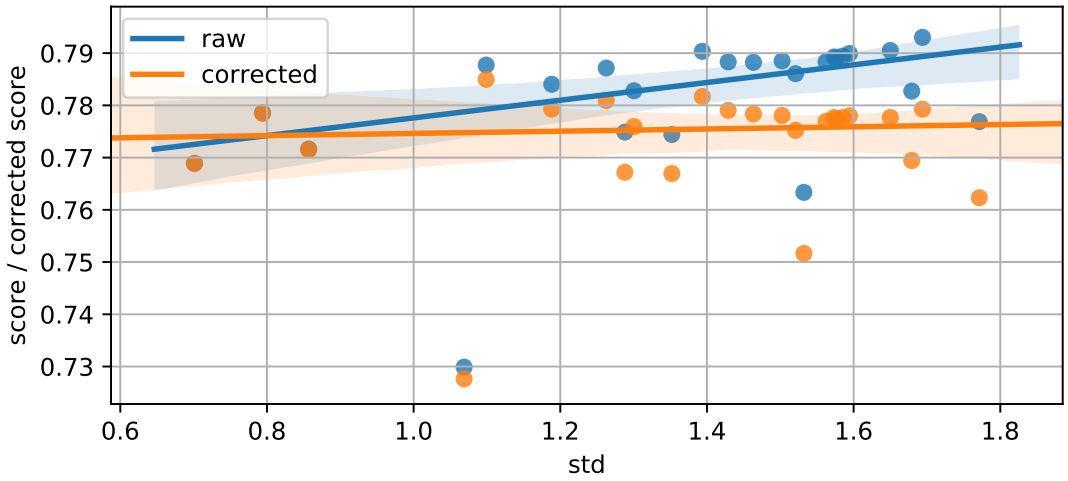


Figure 6.11: Scores (raw and corrected) plotted against mean σ per parameter vector with each point representing a model trained with a different dataset

recognition models it is immensely important to have not only realistic looking faces but faces which encompass the inherent nature of real faces.

Figure 6.12: Training data from “naive uniform 4”

Source	Mode	Corrected score	Corrected rank	Score	Rank
Random	naive normal σ 1	.7715	60.78	.7716	69.04
	naive normal σ 1.5	.7793	52.64	.7840	56.87
	naive normal σ 1.8	.7791	52.96	.7883	52.20
	naive normal σ 2	.7752	57.55	.7861	54.93
	naive uniform 3.5	.7672	64.39	.7749	64.69
	naive uniform 4	.7793	52.13	**.7930**	**46.81**
	naive uniform 5	.7517	78.89	.7633	74.95
	meta normal σ .7 0-25 uniform 6	.7792	52.90	.7855	55.24
	meta normal $\sigma = [.5,2]$	.7813	50.60	.7901	50.40
	normal 8/10 σ 1 uniform 2/10 2.6	.7784	53.83	.7882	52.37
	normal 8/10 σ 2 uniform 2/10 2.6	.7694	63.59	.7827	58.40
	normal σ.7, $beta(\alpha = 3\beta = 15)$ x uniform 6	.7759	56.85	.7828	58.61
p-fit	original data	.7689	63.52	.7689	71.17
	normal σ, μ fitted	.7785	54.65	.7786	61.70
	uniform x 2	.7623	69.93	.7769	63.57
	uniform σ, μ fitted	**.7850**	**45.80**	.7877	52.15
wiki ensemble	CVGAN [2000, 1000, 500] G .5 drop	.7276	92.16	.7299	94.93
	CVGAN [500, 500] .1 noise	.7670	65.78	.7744	66.70
	GMM 1	.7781	54.28	.7885	52.45
	GMM 5	.7769	55.63	.7884	52.58
	GMM 25	.7777	54.66	.7893	51.53
	GMM 100	.7775	54.85	.7891	51.63
	GMM 400	.7780	54.52	.7899	51.17
	GMM 1000	.7777	54.80	.7905	50.35
	GMM 4000	.7776	54.70	.7894	51.38

Table 6.6: Mean scores for predicted parameter vectors. In total models where computed, together with p-fit vectors all predictions are ranked between 1 and 120. Before correction all but one random initialization method performs better or the same as p-fit. Even after correction p-fit doesn't perform better than uniform.

6.4.2 Results

In Figure 6.13 the raw and corrected scores for all models with 95% confidence interval are presented. For raw scores the best predictions, with a recognizable margin to the second place, were generated with training data which was uniformly distributed with -4 to 4. Uniform distributions lower (3.5) and higher (5) performed worse. The model with highest mean corrected score "uniform σ, μ fitted" is based on the data of p-fit. This sampling method is also based on uniform sampling.

The performance of the generative models is mixed. For raw scores the GMMs rank over average but for corrected scores they only rank in the middle. The GANs underperform for both scores and don't yield good results in comparison.

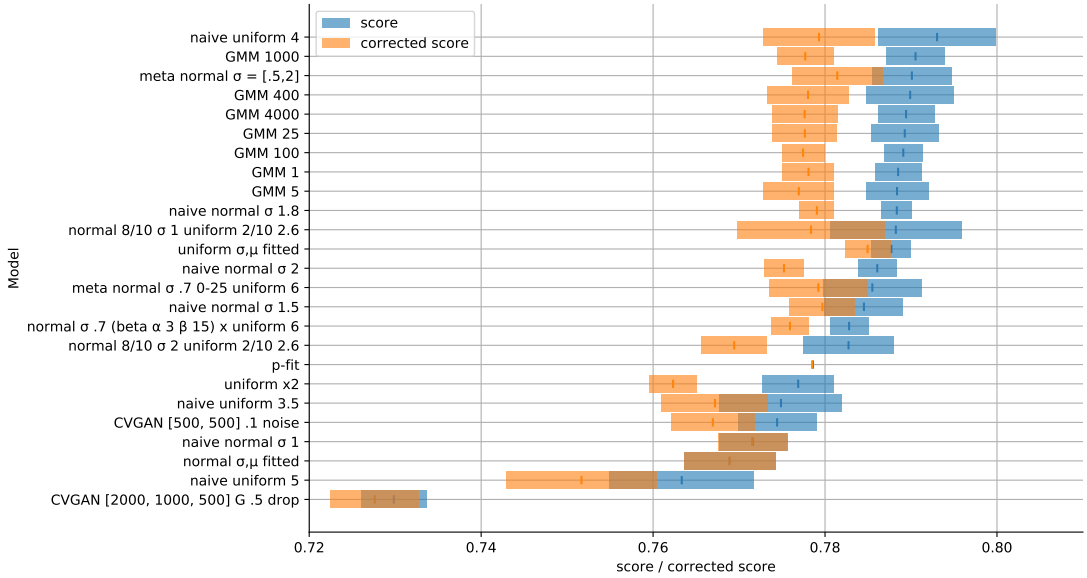


Figure 6.13: Scores (raw and corrected) with upper and lower bound 95% confidence interval

One trend that has been recognized is that for almost all models (with the exception of p-fit datasets) the mean values per parameter for the predictions were qualitatively the same only differentiating in scale (see Figure 6.14). The predictions of models trained with wiki ensemble datasets may be predisposed to these values by the training data, but the models trained with purely random approaches are certainly not. The models trained with pure random data are more fuzzy with mean values than models trained with wiki ensemble which already incorporate these mean values in their training data.

Consistently predicting instances with the same mean value has two connotations. A reason for this behavior is that the set of faces used for the BFM does only show a limited part of the real distribution of faces. The PCA of this distribution therefore does not represent the principal components that would be computed with the real distribution of faces. As previously pointed out, plotting a distribution in the PCA space of a different distribution does yield a different PCA transform. The distribution of the faces used for the prediction most certainly differs from the distribution of the BFM faces.

And a very positive observation that can be seen is that the models, invariant to the training data, do somewhat agree over the distribution of faces that were to be predicted. Even when part of the imdb dataset is predicted the mean values stay almost on the

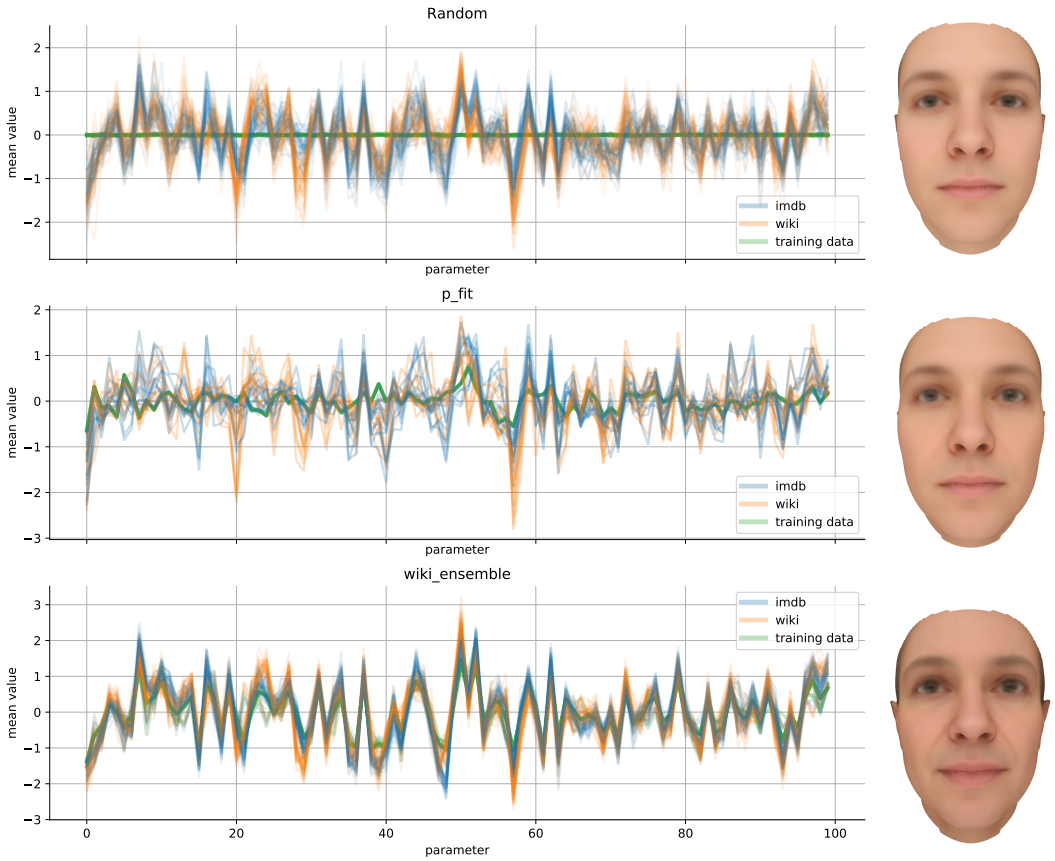


Figure 6.14: mean values per parameter broken down by model, run, source and image dataset used for prediction. On the right side the mean face for every source is displayed.

same level as when part of the wiki dataset is predicted. This indicates that the two hand picked image sets from parts of the wiki and imdb datasets constitute a similar distribution. Because both image sets are larger than 1000 images it is possible that both image sets resemble a representative sample of the real distribution of faces. And this further indicates that the found mean values are probably a better approximation to the real distribution of faces. The mean face for training data based on wiki_ensemble is noticeable older than the mean faces of both other sources, this is a result of the differing age distribution from the wiki and imdb datasets compared to the BFM faces. The mean age for the 3d face of the BFM was between 20 and 30 while for the predicted image sets the mean age is around 40 for this reason it is plausible that the mean face looks older.

In Figure 6.15 scatter plots with per instance standard deviation and corrected score, grouped by data source, are shown. For all models the distribution of scores follows a normal distribution. The distribution of per instance standard deviation is also normal distributed with the exception of "uniform 3.5" and "uniform 5". These two models have a second heap at very low values. This indicates that these models have a hard time predicting some of the images and rather make an average looking model with low standard deviation. This is all the more interesting as "uniform 4" is the best performing model for raw scores and sits between these two datasets. This phenomenon suggests that a good dataset needs larger values while too much caricature like instances are also detrimental. The advantage of uniform datasets is that more higher values are present, in the training the neural net gets a better idea of the effect of a single parameter if it is frequently present with high negative and positive values. This hypothesis is supported by the also relatively high ranking random composite datasets. Two of these datasets ("meta normal $\sigma = [.5, 2]$" and "meta normal .7 0-25 uniform 6", ranking 2nd and 5th) do have high values but also feature many parameters per instance at lower values.

This property might not hold with larger training datasets, because with more data the absolute frequency of higher values even for a normal distributed dataset is high.

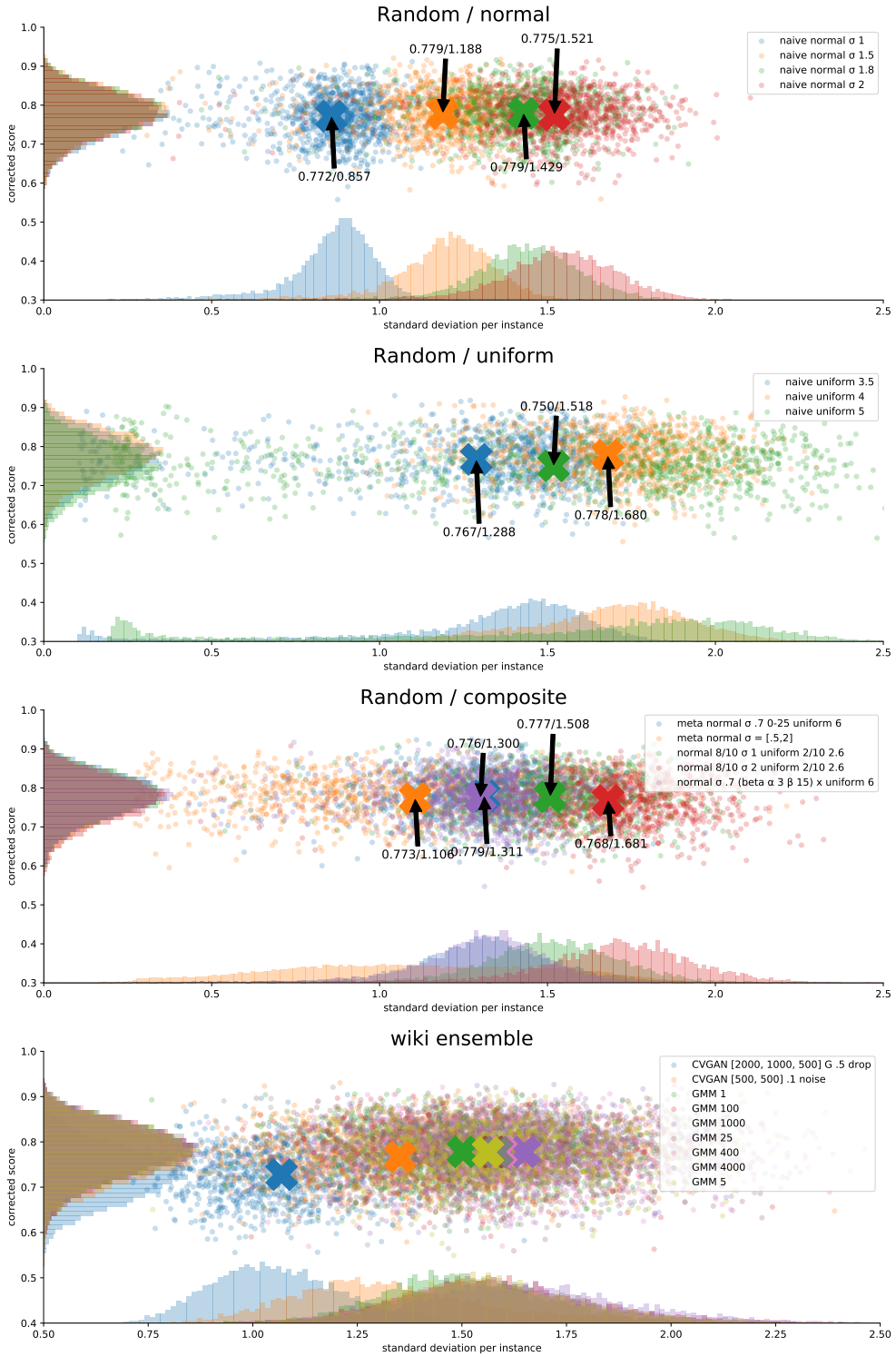


Figure 6.15: Standard deviation per instance vs. corrected score. ✕ marks the centroid of the respective distribution

7 Conclusion

The event space of the BFM has been examined. It was differentiated between valid instances and invalid instances. It has been discovered that purely random sampled instances of the BFM do not resemble a representative sample of real faces. This can be seen most prominently when male and female facial features are examined. The distribution of real faces has a bimodal distribution of these features but a random samples of the BFM only have unimodal distribution. This results in a unclear distinction between male and female instances.

Three different abstraction levels to image generation have been identified. The environment, instance and meta level. To produce good synthetic facial images all levels have to be taken into account.

A CNN was created which regresses parameter vectors for the BFM as predictions for facial images. This net has the WideResNet50 as stem and is appended with two fully connected layers with 200 neurons each. The decision was made to only train the appended layers and exclude the pretrained stem.

On the basis of proper fittings of real images two generative model types, GAN and GMM, have been chosen to produce samples as opposed to purely random generated samples. The initial plan was to supply data from a fitting algorithm as proposed in [35], this was later overturned due to better performance of the BPRNs trained with structureless random datasets. The best predictions from a set of over 40 BPRNs trained with structureless random datasets were chosen to be the basis for the generative models. The GMM was created with an existing framework and modeled with different component sizes.

The GAN architecture was self designed and supports supplementary inputs for age and sex to parametrize the output.

The created GMMs are equal in performance, the different component sizes did not significantly alter the efficacy of the models. The proposed GAN and the GMMs can sample good instances with greater shape and textural emphasis while still maintaining a

reasonable appearance. The GANs also facilitates the creation of learning data controlled for age and gender, which enables the creation of learning data tailored to the need of the application.

The proposed CNN can predict conceivable instances for real images and decently generalizes with purely synthetic data. It performs at least equally good with optimized random learning data as the proposed probabilistic fit and the computation time is immensely reduced. An improvement for future work would be to employ a better performing regression model by either refining the existing model or using a proven reliable model architecture as proposed in [7, 5].

With improved training data a GAN with better knowledge of the face space could be established. Training data may be refined trough an iterative process. It is sensible to dive deeper into the face space of the BFM, instances can be found with greater σ which still resemble real faces. Another opportunity for further research is the exploration of the latent space of the GAN. When relations of latent variables are analyzed an even more fined grained access to sampling of good instances would be possible. It might be for example possible to extract variables that are influence ethnicity of the instances.

The BPRN would most certainly perform better with decoupled pose, illumination and expression. This would partly be possible when these informations would also be supplied by labels.

Research in 3DMM regression with neural nets has only been recently explored. The development in face recognition over the last years are not only funded exceptionally well but pursued by many. Therefore the performance of face recognition is overwhelmingly better than 3DMM regression. The strength of face recognition algorithms to infer the essence of facial identities takes a part of the burden of the shoulders of the 3DMM regression net. Face recognition can be used as an estimator for the quality of 3DMM regression. As proposed in [9, 46] the incorporation of face recognition algorithms in the learning process is reasonable. With direct feedback from face recognition better learning is possible.

The objective of this thesis was to determine if generative models learned on a known valid distribution can sample better training data than random structureless techniques. This question can unfortunately not satisfactorily be answered. The measuring method with face recognition was able to discern bad from good predictions but could not order

higher scoring faces accurately because it preferred faces with exaggerated features over more realistic faces.

The major take aways from this examination were that uniform distributed samples performed better for raw and corrected scores. All models consistently regressed matching mean values per parameter even for different image sets. This implies that the BFM is not based on a representative sample of faces and that the BPRNs were able to identify a similar other distribution in different image sets. The better performance of structureless random data suggests that at least for small datasets realistic looking faces are less relevant than greater diversity and more pronounced features. The failing of generative models may be partly explainable by the omission of valid parts of the face space.

A more promising approach to sampling good training data for similar tasks is the already mentioned incorporation of face recognition and a rendering engine in the model. Such a model will be trained semi-supervised only with a small amount of data supplied at the start to create a sufficient prior. This method will also solve the overfitting issue when training the stem, because training data is generated continuously and will therefore not be repeated.

Bibliography

[1] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision (ECCV)*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, Eds. 2016, vol. 9906 of *LNCS*, pp. 102–118, Springer International Publishing.

[2] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Lüthi, Sandro Schönborn, and Thomas Vetter, "Morphable face models - an open framework," in *13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi'an, China, May 15-19, 2018*. 2018, pp. 75–82, IEEE Computer Society.

[3] Adam Kortylewski, Andreas Schneider, Thomas Gerig, Bernhard Egger, Andreas Morel-Forster, and Thomas Vetter, "Training deep face recognition systems with synthetic data," *CoRR*, vol. abs/1802.05891, 2018.

[4] Elad Richardson, Matan Sela, and Ron Kimmel, "3d face reconstruction by learning from synthetic data," in *Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, October 25-28, 2016*. 2016, pp. 460–469, IEEE Computer Society.

[5] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel, "Learning detailed face reconstruction from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. jul 2017, IEEE.

[6] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li, "Face alignment across large poses: A 3d solution," *CoRR*, vol. abs/1511.07212, 2015.

[7] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni, "Regressing robust and discriminative 3d morphable models with a very deep neural network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. jul 2017, IEEE.

[8] Marcel Piotraschke and Volker Blanz, "Automated 3d face reconstruction from multiple images using quality measures," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. 2016, pp. 3418–3427, IEEE Computer Society.

[9] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman, "Unsupervised training for 3d morphable model regression," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 2018, pp. 8377–8386, IEEE Computer Society.

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 815–823, IEEE Computer Society.

[11] F. J. Anscombe, "Graphs in statistical analysis," *The American Statistician*, vol. 27, no. 1, pp. 17–21, feb 1973.

[12] Bernhard Egger, Dinu Kaufmann, Sandro Schönborn, Volker Roth, and Thomas Vetter, "Copula eigenfaces - semiparametric principal component analysis for facial appearance modeling," in *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 1: GRAPP, Rome, Italy, February 27-29, 2016.*, Nadia Magnenat-Thalmann, Paul Richard, Lars Linsen, Alexandru Telea, Sebastiano Battiato, Francisco H. Imai, and José Braz, Eds. 2016, pp. 50–58, SciTePress.

[13] Tom M. Mitchell, *Machine learning*, McGraw Hill series in computer science. McGraw-Hill, 1997.

[14] Tony Jebara, *Machine Learning: Discriminative and Generative*, Kluwer Academic Publishers, 2004.

[15] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[16] Yoshua Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[17] Stephen I. Gallant, "Perceptron-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 179–191, 1990.

[18] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville, *Deep Learning*, Adaptive computation and machine learning. MIT Press, 2016.

[19] Augustin Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[20] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," 1998.

[21] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[22] Dominik Scherer, Andreas C. Müller, and Sven Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks - ICANN 2010 - 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III*, Konstantinos I. Diamantaras, Wlodek Duch, and Lazaros S. Iliadis, Eds. 2010, vol. 6354 of *Lecture Notes in Computer Science*, pp. 92–101, Springer.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] Karl Pearson F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[25] G. Golub and W. Kahan, "Calculating the singular values and pseudo-inverse of a matrix*," jan 1965, vol. 2, pp. 205–224, Society for Industrial & Applied Mathematics (SIAM).

[26] Frank J. Massey Jr., "The kolmogorov-smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

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

[28] Paul A. Viola and Michael J. Jones, "Robust real-time face detection," in *ICCV*, 2001, p. 747.

[29] Andy Adler and Michael E. Schuckers, "Comparing human and automatic face recognition performance," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 37, no. 5, pp. 1248–1255, 2007.

[30] P. Jonathon Phillips and Alice J. O'Toole, "Comparison of human and computer performance across face recognition experiments," *Image Vision Comput.*, vol. 32, no. 1, pp. 74–85, 2014.

[31] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.

[32] Brendan Klare, Mark J. Burge, Joshua C. Klontz, Richard W. Vorder Bruegge, and Anil K. Jain, "Face recognition performance: Role of demographic information," *IEEE Trans. Information Forensics and Security*, vol. 7, no. 6, pp. 1789–1801, 2012.

[33] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[34] Rasmus Rothe, Radu Timofte, and Luc Van Gool, "Dex: Deep expectation of apparent age from a single image," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015.

[35] A. Schneider, S. Schönborn, B. Egger, L. Frobeen, and T. Vetter, "Efficient global illumination for morphable models," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 3885–3893.

[36] Jan Čech, Vojtěch Franc, Michal Uřičář, and Jiří Matas, "Multi-view facial landmark detection by using a 3d shape model," *Image and Vision Computing*, vol. 47, pp. 60–70, mar 2016.

[37] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter, "Empirically analyzing the effect of dataset biases on deep face recognition systems," *CoRR*, vol. abs/1712.01619, 2017.

[38] Bernhard Egger, Sandro Schönborn, Andreas Schneider, Adam Kortylewski, Andreas Morel-Forster, Clemens Blumer, and Thomas Vetter, "Occlusion-aware 3d morphable models and an illumination prior for face image analysis," *International Journal of Computer Vision*, 2018.

[39] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, Eds., 2014, pp. 487–495.

[40] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," *CoRR*, vol. abs/1605.07146, 2016.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[42] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 2018, pp. 4510–4520, IEEE Computer Society.

[43] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, Eds., 2014, pp. 2672–2680.

[44] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[45] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson, "Understanding neural networks through deep visualization," *CoRR*, vol. abs/1506.06579, 2015.

[46] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker, "Towards large-pose face frontalization in the wild," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2017, pp. 4010–4019, IEEE Computer Society.

Appendix

Following are samples for predicted BFM instances from all models.

Character	Source	Category
a)	real	original image
b)	p-fit	normal
c)		original fit
d)		uniform
e)		uniform $\times$ 2
f)	Random	meta normal σ .7 0-25 uniform 6
g)		normal with $\vec{\sigma}$
h)		naive normal σ 1.5
i)		naive normal σ 1.8
j)		naive normal σ 1
k)		naive normal σ 2
l)		naive uniform 3.5
m)		naive uniform 4
n)		naive uniform 5
o)		normal σ.7, $beta(\alpha = 3, \beta = 15)$ x uniform 6
p)	wiki ensemble	CVGAN2 [500, 500] .1 noise
q)		CVGAN1 [2000, 1000, 500] G .5 drop
r)		GMM 1
s)		GMM 5
t)		GMM 25
u)		GMM 100
v)		GMM 400
w)		GMM 1000
x)		GMM 4000

Table 1: Corresponding Dataset to captioned character

(a)

(b) 0.835

(c) 0.818

(d) 0.842

(e) 0.862

(f) 0.888

(g) 0.840

(h) 0.838

(i) 0.855

(j) 0.828

(k) 0.854

(l) 0.820

(m) 0.855

(n) 0.821

(o) 0.853

(p) 0.775

(q) 0.765

(r) 0.832

(s) 0.823

(t) 0.844

(u) 0.811

(v) 0.839

(w) 0.837

(x) 0.829

(a)

(b) 0.790

(c) 0.761

(d) 0.802

(e) 0.778

(f) 0.789

(g) 0.793

(h) 0.764

(i) 0.770

(j) 0.796

(k) 0.782

(l) 0.770

(m) 0.807

(n) 0.691

(o) 0.787

(p) 0.768

(q) 0.817

(r) 0.811

(s) 0.827

(t) 0.809

(u) 0.788

(v) 0.808

(w) 0.789

(x) 0.776

(a)

(b) 0.801

(c) 0.840

(d) 0.851

(e) 0.894

(f) 0.859

(g) 0.865

(h) 0.848

(i) 0.873

(j) 0.801

(k) 0.850

(l) 0.894

(m) 0.904

(n) 0.877

(o) 0.866

(p) 0.769

(q) 0.826

(r) 0.859

(s) 0.846

(t) 0.858

(u) 0.840

(v) 0.846

(w) 0.844

(x) 0.839

(a)

(b) 0.769

(c) 0.743

(d) 0.777

(e) 0.812

(f) 0.794

(g) 0.774

(h) 0.753

(i) 0.791

(j) 0.770

(k) 0.782

(l) 0.769

(m) 0.839

(n) 0.747

(o) 0.767

(p) 0.743

(q) 0.866

(r) 0.813

(s) 0.800

(t) 0.788

(u) 0.796

(v) 0.823

(w) 0.833

(x) 0.810

(a)

(b) 0.688

(c) 0.730

(d) 0.679

(e) 0.721

(f) 0.791

(g) 0.759

(h) 0.722

(i) 0.715

(j) 0.698

(k) 0.772

(l) 0.718

(m) 0.701

(n) 0.709

(o) 0.726

(p) 0.739

(q) 0.774

(r) 0.737

(s) 0.724

(t) 0.749

(u) 0.749

(v) 0.781

(w) 0.723

(x) 0.719

(a)

(b) 0.744

(c) 0.779

(d) 0.771

(e) 0.792

(f) 0.779

(g) 0.816

(h) 0.786

(i) 0.813

(j) 0.748

(k) 0.816

(l) 0.749

(m) 0.832

(n) 0.694

(o) 0.836

(p) 0.746

(q) 0.849

(r) 0.795

(s) 0.765

(t) 0.771

(u) 0.746

(v) 0.816

(w) 0.831

(x) 0.801

(a)

(b) 0.817

(c) 0.824

(d) 0.839

(e) 0.810

(f) 0.845

(g) 0.829

(h) 0.812

(i) 0.864

(j) 0.826

(k) 0.839

(l) 0.833

(m) 0.832

(n) 0.849

(o) 0.858

(p) 0.842

(q) 0.789

(r) 0.835

(s) 0.845

(t) 0.819

(u) 0.824

(v) 0.873

(w) 0.845

(x) 0.796

(a)

(b) 0.752

(c) 0.726

(d) 0.777

(e) 0.745

(f) 0.754

(g) 0.766

(h) 0.749

(i) 0.780

(j) 0.777

(k) 0.774

(l) 0.764

(m) 0.761

(n) 0.752

(o) 0.773

(p) 0.750

(q) 0.767

(r) 0.798

(s) 0.773

(t) 0.765

(u) 0.770

(v) 0.810

(w) 0.736

(x) 0.803

(a)

(b) 0.779

(c) 0.790

(d) 0.804

(e) 0.754

(f) 0.824

(g) 0.770

(h) 0.818

(i) 0.815

(j) 0.779

(k) 0.739

(l) 0.754

(m) 0.784

(n) 0.753

(o) 0.795

(p) 0.726

(q) 0.754

(r) 0.791

(s) 0.760

(t) 0.773

(u) 0.764

(v) 0.800

(w) 0.774

(x) 0.784

(a)

(b) 0.832

(c) 0.775

(d) 0.835

(e) 0.838

(f) 0.887

(g) 0.894

(h) 0.855

(i) 0.856

(j) 0.827

(k) 0.867

(l) 0.816

(m) 0.835

(n) 0.824

(o) 0.856

(p) 0.768

(q) 0.799

(r) 0.897

(s) 0.892

(t) 0.854

(u) 0.880

(v) 0.857

(w) 0.891

(x) 0.879


(a)


(b) 0.853


(c) 0.851


(d) 0.887


(e) 0.840


(f) 0.867


(g) 0.835


(h) 0.862


(i) 0.863


(j) 0.842


(k) 0.858


(l) 0.788


(m) 0.834


(n) 0.749


(o) 0.868


(p) 0.862


(q) 0.812


(r) 0.831


(s) 0.840


(t) 0.820


(u) 0.814


(v) 0.824


(w) 0.836


(x) 0.806


(a)


(b) 0.867


(c) 0.853


(d) 0.876


(e) 0.871


(f) 0.907


(g) 0.876


(h) 0.877


(i) 0.880


(j) 0.896


(k) 0.883


(l) 0.866


(m) 0.880


(n) 0.873


(o) 0.881


(p) 0.858

(q) 0.864

(r) 0.883

(s) 0.880

(t) 0.903

(u) 0.885

(v) 0.879

(w) 0.918

(x) 0.916

(a)

(b) 0.833

(c) 0.804

(d) 0.822

(e) 0.822

(f) 0.845

(g) 0.830

(h) 0.841

(i) 0.820

(j) 0.848

(k) 0.830

(l) 0.805

(m) 0.822

(n) 0.777

(o) 0.833

(p) 0.803

(q) 0.809

(r) 0.825

(s) 0.813

(t) 0.821

(u) 0.825

(v) 0.787

(w) 0.810

(x) 0.845

(a)

(b) 0.741

(c) 0.766

(d) 0.790

(e) 0.693

(f) 0.747

(g) 0.778

(h) 0.760

(i) 0.738

(j) 0.759

(k) 0.751

(l) 0.702

(m) 0.828

(n) 0.711

(o) 0.750

(p) 0.709

(q) 0.785

(r) 0.841

(s) 0.782

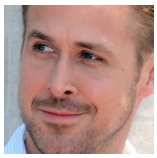
(t) 0.767

(u) 0.811

(v) 0.819

(w) 0.764

(x) 0.787



(a)

(b) 0.774

(c) 0.778

(d) 0.790

(e) 0.826

(f) 0.812

(g) 0.807

(h) 0.789

(i) 0.807

(j) 0.794

(k) 0.804

(l) 0.801

(m) 0.810

(n) 0.794

(o) 0.779

(p) 0.703

(q) 0.792

(r) 0.803

(s) 0.817

(t) 0.788

(u) 0.805

(v) 0.787

(w) 0.813

(x) 0.797

(a)

(b) 0.825

(c) 0.731

(d) 0.863

(e) 0.861

(f) 0.859

(g) 0.856

(h) 0.871

(i) 0.873

(j) 0.829

(k) 0.876

(l) 0.879

(m) 0.859

(n) 0.852

(o) 0.842

(p) 0.817

(q) 0.784

(r) 0.858

(s) 0.853

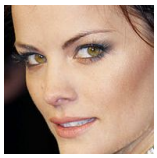
(t) 0.851

(u) 0.859

(v) 0.848

(w) 0.835

(x) 0.841



(a)

(b) 0.756

(c) 0.760

(d) 0.802

(e) 0.746

(f) 0.792

(g) 0.759

(h) 0.771

(i) 0.753

(j) 0.740

(k) 0.792

(l) 0.783

(m) 0.763

(n) 0.759

(o) 0.769

(p) 0.740

(q) 0.767

(r) 0.747

(s) 0.761

(t) 0.771

(u) 0.783

(v) 0.776

(w) 0.775

(x) 0.766

(a)

(b) 0.839

(c) 0.764

(d) 0.830

(e) 0.743

(f) 0.825

(g) 0.815

(h) 0.825

(i) 0.805

(j) 0.845

(k) 0.800

(l) 0.798

(m) 0.805

(n) 0.785

(o) 0.822

(p) 0.766

(q) 0.829

(r) 0.783

(s) 0.813

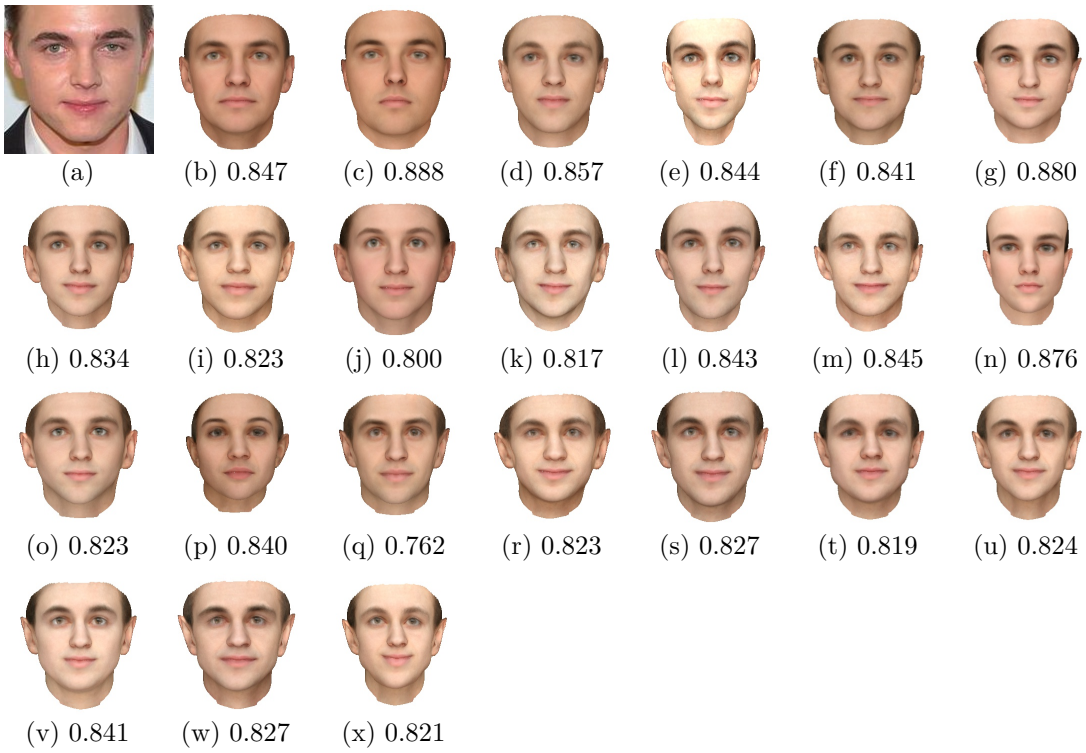
(t) 0.825

(u) 0.791

(v) 0.773

(w) 0.792

(x) 0.785

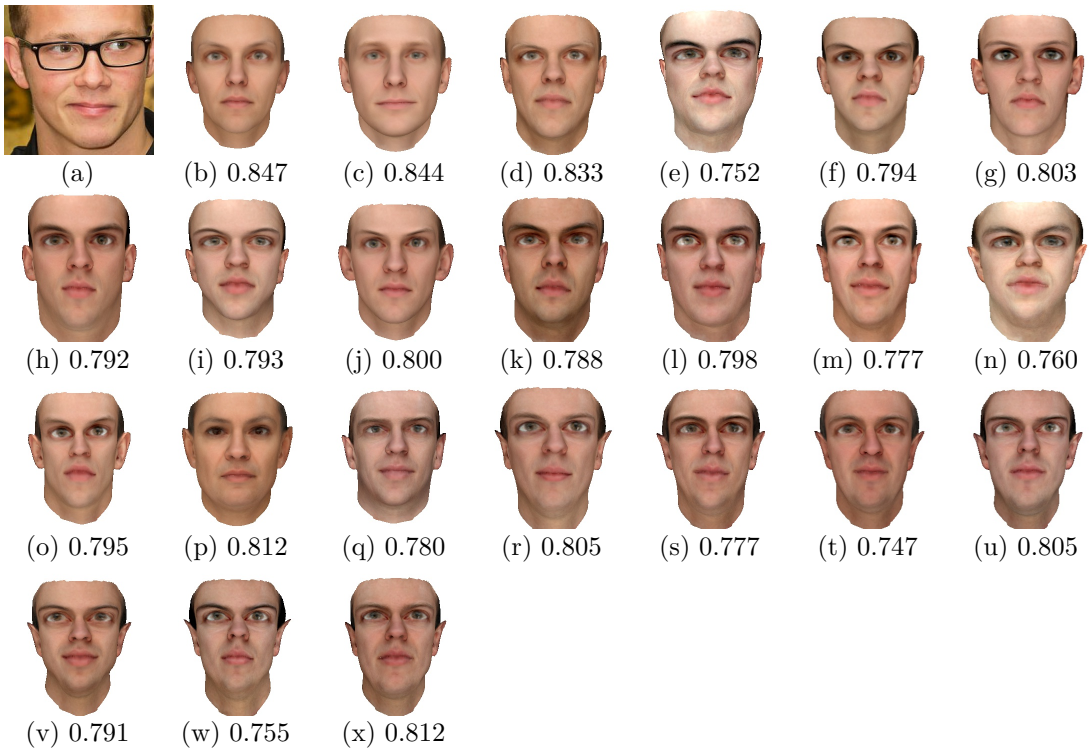


(a) (b) 0.847 (c) 0.888 (d) 0.857 (e) 0.844 (f) 0.841 (g) 0.880

(h) 0.834 (i) 0.823 (j) 0.800 (k) 0.817 (l) 0.843 (m) 0.845 (n) 0.876

(o) 0.823 (p) 0.840 (q) 0.762 (r) 0.823 (s) 0.827 (t) 0.819 (u) 0.824

(v) 0.841 (w) 0.827 (x) 0.821



(a) (b) 0.847 (c) 0.844 (d) 0.833 (e) 0.752 (f) 0.794 (g) 0.803

(h) 0.792 (i) 0.793 (j) 0.800 (k) 0.788 (l) 0.798 (m) 0.777 (n) 0.760

(o) 0.795 (p) 0.812 (q) 0.780 (r) 0.805 (s) 0.777 (t) 0.747 (u) 0.805

(v) 0.791 (w) 0.755 (x) 0.812

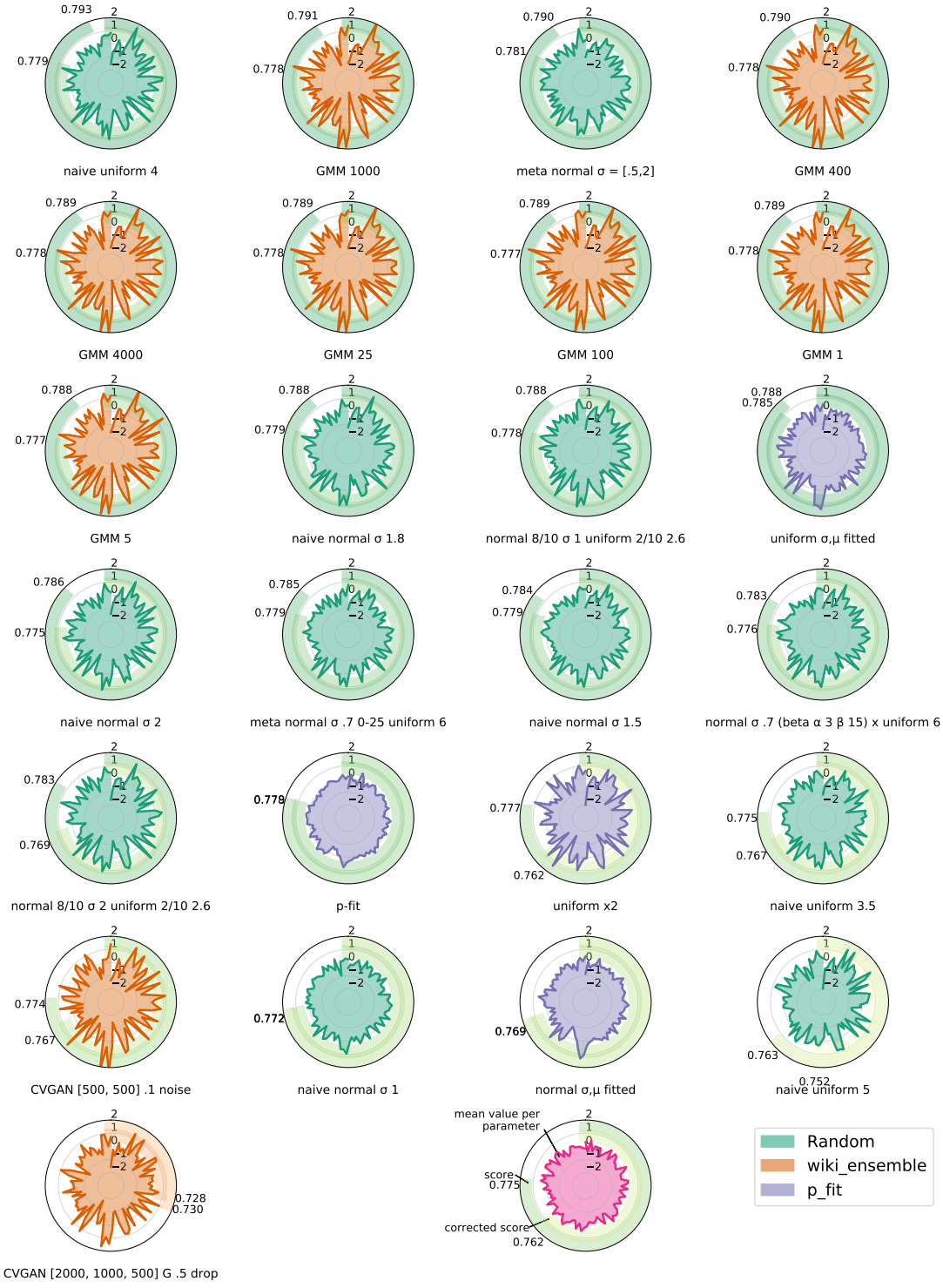


Figure 21: All datasets compared with score, corrected score and mean value for every parameter. Parameters are spread counter-clockwise from 0 to 99 with shape on the left and texture on the right side.

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 11. Januar 2019 Jan Scholz