Robust Discrimination and Generation of Faces using Compact, Disentangled Embeddings

Björn Browatzki
Korea University
Seoul
browatbn@korea.ac.org

Christian Wallraven
Korea University
Seoul
wallraven@korea.ac.kr

Abstract

Current solutions to discriminative and generative tasks in computer vision exist separately and often lack interpretability and explainability. Using faces as our application domain, here we present an architecture that is based around two core ideas that address these issues: first, our framework learns an unsupervised, low-dimensional embedding of faces using an adversarial autoencoder that is able to synthesize high-quality face images. Second, a supervised disentanglement splits the low-dimensional embedding vector into four sub-vectors, each of which contains separated information about one of four major face attributes (pose, identity, expression, and style) that can be used both for discriminative tasks and for manipulating all four attributes in an explicit manner. The resulting architecture achieves state-of-the-art image quality, good discrimination and face retrieval results on each of the four attributes, and supports various face editing tasks using a face representation of only 99 dimensions. Finally, we apply the architecture's robust image synthesis capabilities to visually debug label-quality issues in an existing face dataset.

1. Introduction

Deep learning algorithms have enabled impressive performance for image categorization [16] or for recognition of faces in very large databases [7]. In parallel, developments in generative approaches have demonstrated that deep neural networks also are able to synthesize highly realistic output, for example, for scenes [29] or for high-resolution faces [20]. Both discriminative and generative techniques, however, have come under scrutiny recently for being vulnerable to adversarial attacks and for a lack of interpretability [24], leading to a push to develop more "explainable" networks [3, 44]. Importantly, both types of techniques coexist separately but are rarely brought together in a common framework - such a framework would allow, for example, to discriminate a face but it would also be able to provide an explanation of its decision through its generation and editing capabilities.

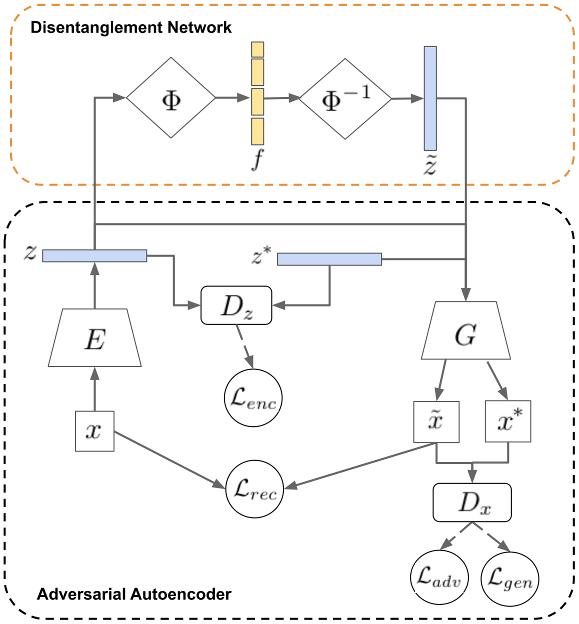

Figure 1: System overview with an unsupervised autoencoder and an attribute-disentangling network. For notation, see Sec.3.

In the present paper, we provide such a framework that provides generative and discriminative paths in one common representation and demonstrate it in the application domain of faces (although our approach works with all types of data). The core idea of our framework is to create a compact, low-dimensional embedding of faces that can be used both for generation and for discrimination (akin to the "face space" conceptual framework from cognitive neuroscience [39]). In order to generate such a low-dimensional embedding, we train an *unsupervised* adversarial autoencoder on a large number of faces containing a lot of variation in face attributes. In our case, the autoencoder yields a 99-dimensional embedding vector that is able to reconstruct the training faces and to generate images of high quality. This vector, however, will most likely contain information about different face attributes (identity, pose, expression, gender, age, etc.) in a highly entangled fashion. In the next step, we therefore add a *supervised disentanglement* step that splits the 99 dimensions into attribute-specific, separate sub-spaces. Owing to the availability of large-scale, annotated face databases, we can encode any number of different face attributes - here, we focus on four ”major” face attributes of identity, expression, pose, and style. Importantly, the disentanglement step ensures that the information in the 99 dimensions is separated *and useful for discrimination*, but it also makes use of the generative aspects by creating its own augmented training images that are fed into the training cycle. Our network architecture (see Fig.1) uses a number of cross-checks to ensure consistency of the feature vectors and the resulting, low-dimensional face embedding. With this architecture, we can

- create realistic-looking faces within standard GAN-like applications
- recognize identities, expression, and pose with, especially, expressions reaching state-of-the-art performance on a challenging, large dataset
- conduct attribute-specific face editing
- explore failure cases of discrimination by means of the robust, generative capabilities to check ”where things went wrong and why”

with a low-dimensional embedding of just 99 dimensions.

2. Related Work

Discriminative methods based on convolutional neural networks (CNNs) have shown impressive results in a number of areas in computer vision, defining state-of-the-art performance in face detection [35], verification [7], attribute [11] and expression recognition [40] (for a combined multi-task approach, see [32]. On the other side, generative adversarial networks (GANs) [9] are able to generate data from random noise through adversarial training enabling super-resolution [25, 43], face synthesis [19], and image-to-image translation [42,46]. Since GANs *per se* do not offer inference capabilities, a number of generative methods have been proposed that build upon an autoencoder architecture [4,15] to map input images into a latent space which can also be sampled from to generate new images [23,26].

In the context of face processing, a number of GAN network architectures were proposed that enable realistic reconstructions of face images combined with the generation of random faces [13,33,38]. However, the encoding of input images does not perform well in regard to preserving facial attributes, such as an individual’s expression or identity. To overcome this issue, approaches have been studied that condition the encoding and generation process on specific facial attributes. For example, [45] propose a conditional adversarial autoencoder (CAAE) that is able to generate faces by controlling the observed age of the face. Similar methods have been developed to generate additional training data for discriminative face processing tasks. For example, [18] or [37] use pose information to improve face verification performance and [8] exploit the idea of a face space defined by a 3D model to generate faces conditioned on expression, pose, and identity. Generative adversarial methods have also targeted facial expressions analysis. In [36], for example, fiducial points are employed to control facial expression synthesis. Gu et al. [10] modify facial expression in training images to increase the accuracy of a facial expression classifier.

Recent work in generative methods have attempted to disentangle facial attributes in a more controlled fashion. In [45], for example, pose and expression are separated from identity for independent face synthesis. An approach that is similar to ours in spirit is [34] in which an autoencoder learns the disentanglement into an identity-distilling feature and an identity-dispelling feature. One feature can be used for discriminative identity recognition, whereas the remaining feature can be modified independently to control other attributes in a generated face. Importantly, existing work like [34] so far has only dealt with one or two explicit facial attributes. Our method extends this by disentangling *four major attributes* (pose, identity, expression, and style) in a face image at the same time. It also adds two important architectural elements: first, we separate disentanglement from face space learning, thus allowing training of the latter on very large datasets of unlabeled data in an unsupervised fashion. Second, through the incorporation of cycle-consistency into the disentanglement process, we are able to separate multiple features in parallel without the need for a feature-wise information distilling/dispelling scheme as used in [34] which allows for an integrated formulation that can also be extended with additional attributes.

3. Methods

In this section we describe the two components of our approach. First, an adversarial autoencoder learns an unsupervised feature representation $z \in \mathbb{R}^d$ of face images. A subsequent supervised process then splits this feature vector into a disentangled representation $f = (f_p, f_{id}, f_e, f_s)$ consisting of an independent encoding for *pose, identity, expression*, and *style*, respectively. Style serves as a residual feature capturing aspects in the image that are not attributed to either pose, identity, or expression.

3.1. Unsupervised adversarial autoencoder

Fig.1 shows the architecture of the autoencoder that is used to encode facial images into a feature vector representation. Given an input image x, the encoder network $E(x)$ produces a feature vector $z \in \mathbb{R}^d$. The decoder $G(z)$ projects z back into image space: $\tilde{x} = G(E(x))$. The encoding and decoding process is guided by a set of loss functions that ensure the faithful reconstruction of input images as well as the learning of a smooth manifold of faces.

Reconstruction loss : An accurate representation of input faces is critical for subsequent classification and editing tasks. This is achieved through an $\ell 1$ reconstruction loss that penalizes pixel errors:

$$\mathcal{L}_{rec}(E, G) = \mathbb{E}_{x \sim p(x)}[\|x - G(E(x))\|_1] \tag{1}$$

Adversarial feature loss: Following [9], a discriminator D_z imposes the prior distribution $p^* \sim p(z)$ on z. This forces the generator E to produce a continuous embedding space that can be sampled to generate new images.

$$\begin{aligned}\mathcal{L}_{enc}(E, D_z) = \mathbb{E}_{z^* \sim p(z)}&[\log D_z(z^*)] \\ + \mathbb{E}_{x \sim p(x)}&[\log{(1 - D_z(E(x)))}]\end{aligned} \tag{2}$$

Adversarial image loss: Autoencoders trained with image reconstruction losses typically suffer from creating blurry images. We observed the same behavior leading to accurate image reconstructions in terms of pixel differences, yet smoothing out finer details such as wrinkles. However, for tasks such as expression recognition higher-frequency information is vital [41]. Hence, an adversarial loss,

$$\begin{aligned}\mathcal{L}_{adv}(E, G, D_x) = \mathbb{E}_{x \sim p(x)}&[\log D_x(x)] \\ + \mathbb{E}_{x \sim p(x)}&[\log(1 - D_x(G(E(x))))],\end{aligned} \tag{3}$$

is added where encoder E and decoder G are optimized to auto-encode images $\tilde{x} = G(E(x))$ that look similar to the input images x from the training set. The discriminator D_x tries to distinguish real images from their reconstructions $\tilde{x}$. D_x can only be fooled if the same amount of detail is present in the auto-encoded face as in the original image.

Generative image loss: The adversarial image loss has the drawback that it only operates on images from the training set. It does not encourage the generator to create realistic face images from the latent code z. The generator is thus prone to overfitting training images. We therefore add the adversarial loss of a Generative Autoencoder Network:

$$\begin{aligned}\mathcal{L}_{gen}(G, D_x) = \mathbb{E}_{x \sim p(x)}&[\log D_x(x)] \\ + \mathbb{E}_{z^* \sim p(z)}&[\log{(1 - D_x(G(z^*)))}]\end{aligned} \tag{4}$$

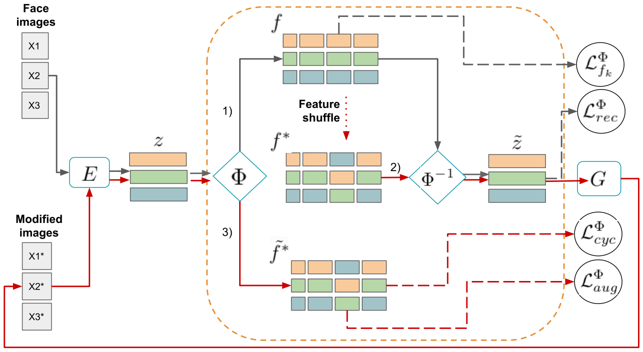

Figure 2: Data flow through the disentanglement procedure. Stages 1), 2) and 3) are referenced in the text.

This encourages the generator to produce realistic images for any $z \in p(z)$, not only instances of z that are created by the encoder.

Full autoencoder objective: The full objective of our adversarial autoencoder is given by:

$$\begin{aligned}\min_{E,G} \max_{D_z, D_x} \mathcal{L}_{AE}(E, G, D_z, D_x) = \\ \lambda_{AE}(\mathcal{L}_{rec}(E, G) + \mathcal{L}_{adv}(E, G, D_x)) \\ + \mathcal{L}_{enc}(E, D_z) + \mathcal{L}_{gen}(G, D_x)\end{aligned} \tag{5}$$

where $\lambda_{AE} = 0.9$ weights training image reconstruction versus quality of random image generation.

3.2. Disentangling facial components

After the autoencoder has learned a general face encoding z, the aim is to disentangle z into a new latent variable f which is composed of sub-vectors representing individual facial attributes. As depicted in Fig.1, the disentanglement consists of an autoencoding scheme with z supplied to a network $\Phi(z) \rightarrow f$ and its output f (Fig.2 stage 1)) returned to the inverse network $\Phi^{-1}(f) \rightarrow \tilde{z}$ to reconstruct the original vector z.

3.3. Encoding reconstruction loss

The two feature space transformations should lead to an identity cycle with $\tilde{z} \approx \Phi^{-1}(\Phi(z))$. This is enforced through the reconstruction loss $\mathcal{L}^{\Phi}_{rec}$ as defined by:

$$\mathcal{L}^{\Phi}_{rec} = \mathbb{E}_{z \sim p_{data}(z)}[\left\|z - \Phi^{-1}(\Phi(z))\right\|_1] \tag{6}$$

We chose the $\ell 1$ distance here since its linear output range balances this value against the other loss terms.

3.4. Discriminative task loss

The disentanglement is driven by supervised task-specific loss terms $\mathcal{L}^{\Phi}_{f_k}$ that are calculated by evaluating feature distance functions $\mathcal{F}_{||}$ with $k \in p, id, e, s$ for pose, identity, expression, and style.

$$\mathcal{L}^{\Phi}_{f_k} = \mathbb{E}_f[\mathcal{F}_k(f_k, y_k)], \tag{7}$$

where y_k denotes ground truth labels. Training images during the disentanglement process are taken from different datasets that contain varying types of annotations of pose, identity, expression, and style. We obtain ground truth identities y_{id} from datasets for face identification and expression annotations from FER datasets [27]. Head poses y_p are extracted for all datasets using a face detection algorithm [1]. Lastly, we collect information on image and face style by considering variances within and across video clips [28]. More details and feature comparison metrics are given in Sec.3.7.

3.5. Attribute cycle-consistency loss

Through supervised encoding following Eq.(7), information regarding a certain facial factor k is distilled into a attribute vector f_k. For a disentangled representation, however, we need to ensure that no information about this attribute is present in the remaining vectors f_j with $j \neq k$.

This is achieved using a cycle-consistency loss on a modified attribute vector $f^*_k = \pi(f_k)$. Other components of the feature vector $f_i, i \neq j$ are kept unchanged. The new vector f^* is passed to the reconstruction pipeline $G(\Phi^{-1}(f)) \to x$ to produce a novel generated face image x^* (Fig.2 stage 2)). This image differs from the original input image by the edits carried out in f_k. Image x^* is then returned to the encoding pipeline $\Phi(E(x)) \to f$ to obtain a reconstructed feature vector $\tilde{f}^*$ (Fig.2 stage 3)). Loss $\mathcal{L}^{\Phi}_{cyc}$ penalizes errors during this process:

$$\mathcal{L}^{\Phi}_{cyc} = \mathbb{E}_z[\|\pi(f) - \mathcal{I}(\pi(f)\|_1] \tag{8}$$

In Eq.(8) $\mathcal{I}(f)$ denotes an identity function representing one full pass through autoencoder and disentanglement,

$$\mathcal{I}(f) = \Phi(E(G(\Phi^{-1}(f)))), \tag{9}$$

and $\pi(f)$ is a feature editing function. To guarantee that all edited attribute vectors $\tilde{f}_k$ are valid encodings, π does not create new vectors but performs a random permutation across a training batch.

If attributes are disentangled and independent from one another, Eq.(8) will return low values. Consider for example an image with an angry expression. If the encoded angry feature is replaced by an encoded happy expression, the generated new image will only differ from the input image by a modified facial expression. If this image is encoded again, the resulting attribute vector should be identical to the attribute vector of the original (happy) image with the replaced (angry) expression vector. If disentanglement fails, this is due to one of three cases:

- Expression data is present in other attribute vectors: This leads to a generated image not depicting a purely angry expression. Encoding would reflect this in producing a non-angry expression vector.
- Other information (*e.g.* identity) is present in the expression vector: This leads to more information being transfered during the feature shuffle than only expression. The generated image will exhibit a change in identity and the resulting identity code will differ from the code of the original image.
- No relevant information is contained in the expression vector. Since generated images will not be affected by the content of the vector, also encoded feature vectors will not be different from original ones. Eq.(8) would in fact output a low loss. However, if this were the case the task loss in Eq.(7) would return very high errors.

The flow of data through our networks to calculate the cycle-consistency loss is shown in Fig.2 by the red arrows.

3.6. Attribute cycle-augmentation loss

Through the shuffling process, novel face images are created. Yet, the attribute cycle-consistency loss (8) does not consider the correctness of the recognized attributes on these images. On the other hand, the task loss (7) only operates on real images from the training set. To take full advantage of the cycle process, we therefore combine (8) and (7) into a new loss $\mathcal{L}^{\Phi}_{aug}$:

$$\mathcal{L}^{\Phi}_{aug} = \mathcal{F}_k(\mathcal{I}(\pi(f))_k, \pi(y_k)) \tag{10}$$

Here the feature shuffling function π produces the same output for features f and labels y. By requiring that the encoded attributes from the generated images after feature shuffling match the ground truth labels we perform implicit data-augmentation. The generated images are not part of the training set, however, although it is possible to assign labels for supervised training. Consistency in regard to image content and label data is assured by the attribute cycle-consistency loss $\mathcal{L}^{\Phi}_{cyc}$ as illustrated in Sec.3.5

3.7. Supervised feature losses

For the attribute of **Pose**, the supervised loss consists of the differences for yaw, pitch, roll $y_p = (\varphi, \theta, \psi)$

$$\mathcal{F}_p = \|y_p - f_p\|_1 \tag{11}$$

3.7.1 Attribute embedding via triplet loss

Additional attributes in our network are analyzed via triplet losses that preserve relative distances in the embedding with a general equation for the loss of:

$$\begin{aligned} tl_k(f, f^+_k, f^-_k) = \max(0, &\left\|f_{att} - f^+_k\right\|^2_2 \\ &- \left\|f_k - f^-_{att}\right\|^2_2 + \delta_k) \end{aligned} \tag{12}$$

where k indexes the attribute, and for a given feature f we select f_k^+ from the same class as f and f_k^- from a different class as f.

Identity tl_{id}: To calculate $\mathcal{F}_{id}$ we create a triplet loss by regarding an image of the same identity as a positive match and as a negative match if from any other identity. We create triplets within batches. For each image in a training batch a random positive and a random negative feature is picked from the same batch.

Expression tl_e: To encode expression information we need to a way to distinguish face images based on the depicted expression. Emotion category labels from datasets such as [27] could be used but this would disregard the fact that images of the same emotion can display very different facial expressions [21]. Furthermore, as discussed below, category annotations are sometimes inconsistent and the encoding has to deal with ambiguous information. To mitigate these issues, we additionally consider annotated valence and arousal scores from the AffectNet dataset. Given an expression label $y_e = (v, a, l)$, with valence $v \in [0, 1]$, arousal $a \in [0, 1]$, and emotion label $l \in N^m$ we measure the expression distance between samples as:

$$dist(y_e^i, y_e^j) = \left\|(v,a)^i - (v,a)^j\right\|_2^2 + \lambda_l \delta_{l^i, l^j} \tag{13}$$

where $\delta_{x,y}$ refers to the Kronecker delta returning 1 for $x = y$ and 0 otherwise. $\lambda_l = 0.25$ controls the weight of the expression category information.

For each sample i in a batch of length n we order all other samples in the batch by their distance to i according to Eq.(13). The feature vector of the closest sample is selected as positive feature f_e^+ and a random sample in the interval $[k, n)$ of ordered samples is selected as negative feature f_e^-.

Style tl_s: The style attribute is supposed to encode image content that is not represented by any of the other components. This includes background appearance, illumination as well as attributes and object in the face that are not (necessarily) linked to an identity such as sun glasses, hats or beards. To learn this space of style information we need to find images that are constant within this component but vary across others. We chose video clips of VoxCeleb with annotated identity labels to create triplets for learning an embedding. Let f_s^i be from a video clip v_a, we find a positive match by selecting an image from a different video clip v_b but depicting the same person.

3.7.2 Full disentanglement objective

The full objective for the optimization of the disentanglement networks Φ, Φ^{-1} is stated as:

$$\min_{\Phi,\Phi^{-1}} \mathcal{L}_\Phi = \lambda_{rec}\mathcal{L}_{rec}^\Phi + \mathcal{L}_{f_k}^\Phi + \mathcal{L}_{cyc}^\Phi + \mathcal{L}_{aug}^\Phi \tag{14}$$

Training of these networks can be done using a fixed version of the autoencoder. Since Φ and Φ^{-1} only perform transformations of low-dimensional feature vectors, these networks contain few layers and parameters. Training runs very efficiently with the biggest contribution to computation time being the feed-forward process through the autoencoder for the cycle losses.

3.8. Joint training

The adversarial autoencoder is trained fully unsupervised without any task-specific feedback. In contrast, the disentanglement is limited by the descriptive power of the encoded feature vectors z. Through training both modules jointly, these feature encodings can be adjusted to better capture content required by subsequent tasks. Importantly, our experiments have shown, however, that training all networks jointly *from scratch* is infeasible due to the high number of competing loss terms. To overcome this issue, we conduct joint training only once unsupervised training of the autoencoder and subsequent, supervised training of the disentanglement has reached convergence and losses stabilize. The training objective for joint training is then formalized as:

$$\min_{E,G,\Phi,\Phi^{-1}} \max_{D_z,D_x} \mathcal{L}_{AE}(E, G, D_z, D_x) + \mathcal{L}_\Phi(\Phi, \Phi^{-1})$$

3.9. Multi-dataset and multi-feature training

To create triplets for identity and style we iterate over training data in macro-batches. A macro-batch contains 20 persons for identity training, and 5 persons with 4 video clips each for style. Disentanglement needs to be trained on multiple datasets in parallel since no single dataset contains labels for all attributes. In [6] a scheme was suggested that joins samples from multiple sources into joint batches and maintains indexing vectors to associate samples with source domains. In our approach this is not feasible since we need to iterate in macro-batches. Mini-batchsize would have to be prohibitively large to guarantee the presence of enough identities/clips to find positive and negative triplet pairs.

4. Implementation

4.1. Network architecture

The entire system consists of six networks: four networks for the autoencoder (encoder E, generator G, latent space discriminator D_z, real/fake image discriminator D_x), and two networks for the disentanglement (Φ and Φ^-1).

Encoder/Generator The backbone of our system is a ResNet18 [12] for E and an inverted ResNet18 for G. The inverted ResNet has the first convolutional layers in each block replaced with a 4×4 deconvolutional layer. Encoder input size is 128×128, output size a 99-dimensional vector (similar to the 100d input size of DCGAN [31]). Input and output sizes are swapped for the generator.

Discriminators D_z is a simple network with 3 fully-connected layers with 1000 dimensions each. We use batch normalization, dropout ($p = 0.2$) and the ReLu activation function. For D_x we adopt the architecture of the DCGAN discriminator [31].

Disentanglement networks Networks Φ and Φ^{-1} consist of two fully-connected layers with 1000 dimensions. We do not use batch normalization or dropout layers, as we did not observe performance differences by adding these layers. Input and output sizes for both networks have the same dimensionality as the output size of the encoder E (99d). Attribute features vectors have dimensionalities 3,32,32,32 for f_p, f_{id}, f_e, f_s respectively.

4.2. Image preprocessing

We perform face detection, fiducial point detection, and head pose estimation using OpenFace2.0 [1]. Images from all datasets are rotation aligned and cropped using the extracted fiducial points. Finally, histogram equalization is applied to the extracted crops.

4.3. Training details

During training we mirror images with a probability of 50%. We do not perform other forms of data augmentation. We train the autoencoder with a constant learning rate of learning rate of $5 * 10^{-}5$, and employ the Adam optimizer [22] ($\beta_1 = 0.0, \beta_2 = 0.999$) for parameter update. For disentanglement training the learning rate is set to $10^{-}4$ with Adam $\beta_1 = 0.9/\beta_2 = 0.999$. Batch size is 100 images for both stages.

5. Experiments[1]

5.1. Datasets

CelebA consists of 200K images of celebrities with annotated facial attributes (160K images for training, 19k for validation and 19k for testing) and is commonly used to evaluate generative networks. We employ CelebA for identification training as well. For the larger dataset, we use VGGFace2 [5], from which we take 1M faces for training and 170k for testing. We use LFW [17] for identity testing with 1000 face pairs, AffectNet [27] with eight annotated facial expressions and valence/arousal ratings containing 288K training and 4K validation images for expression training, and VoxCeleb [28] with video clips of 1k different speakers for training of ”style”.

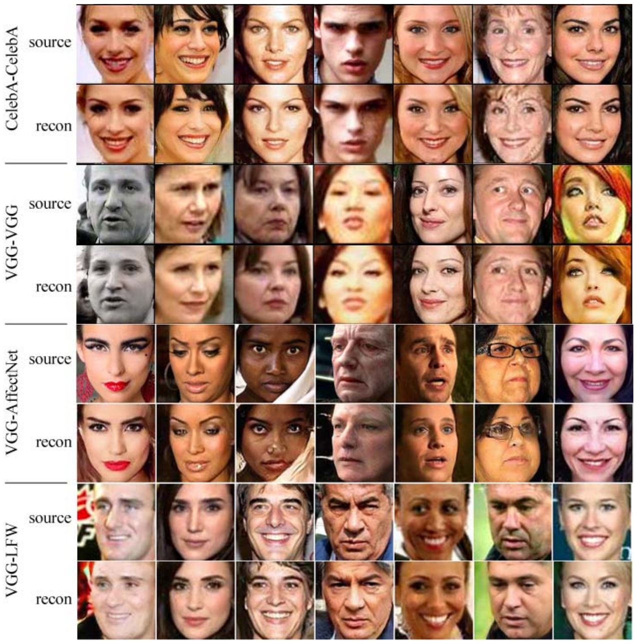

Figure 3: Image reconstructions for CelebA, VGG, AffectNet, and LFW test sets.

	FID	RMSE			
Training set		CelebA	AffectNet	VGG	LFW
CelebA (160K)	17.79	22.87	23.64	22.87	21.23
VGGFace2 (1M)	18.01	19.56	21.78	21.21	20.41

Table 1: Comparison of reconstruction errors (RMSE) for models trained of CelebA and VGGFace2 across different test sets containing unseen images. Image size is 128 × 128.

5.2. Autoencoder training

5.2.1 Reconstruction and random face generation

To evaluate the performance of our adversarial autoencoder we first train a model on the training set of CelebA. We aim for high reconstruction accuracy on test images (shown as CelebA-CelebA in Fig.3) while still being able to generate visually appealing images with a high degree of variation (see supplementary materials).We measure reconstruction accuracy by calculating RMSE values on the CelebA test set. To evaluate generated faces we employ the Fréchet Inception Distance (FID) [14] that measures the distance between two distributions. We extract 10k real images from the CelebA training set and generate 10k random images for comparison. Reconstruction errors are listed in Tab.1.

5.2.2 Full training on VGGFace2

Since we want to capture as much face variation as possible, we next train a new autoencoder on VGGFace2, a much larger database of faces. Reconstructions on this

[1] For experiments on loss terms, parameter tuning, and visualization of the embedding, see supplementary materials.

Training set	f_p	f_e	f_s	$f_{\neg id}$	f_{id}
CelebA	55.8	64.9	62.7	67.5	80.3
CelebA (joint)	53.3	64.3	62.2	66.3	**85.4**
VGG (joint)	53.6	63.7	61.3	64.4	**95.8**

Table 2: Face verification accuracy evaluated on LFW for separate and joint training. Chance level is 50%.

	neut	happ	sad	surp	fear	disg	ang	cont	**AVG**
ATL [40]	0.86	0.96	0.89	0.89	0.90	0.84	0.88	0.83	**0.88**
RN [12]	0.82	0.95	0.89	0.85	0.91	0.87	0.86	0.88	0.88
VGG [30]	0.76	0.92	0.81	0.81	0.82	0.81	0.77	0.82	0.85
f_p	0.53	0.58	0.51	0.54	0.58	0.56	0.56	0.58	0.55
f_{id}	0.65	0.73	0.69	0.74	0.72	0.69	0.73	0.67	0.70
f_s	0.61	0.68	0.63	0.66	0.67	0.61	0.61	0.62	0.64
$f_{\neg e}$	0.66	0.78	0.73	0.76	0.77	0.72	0.76	0.71	0.74
f_e	0.76	0.92	0.80	0.82	0.81	0.78	0.81	0.81	0.81
Joint f_p	0.57	0.59	0.54	0.55	0.66	0.56	0.54	0.61	0.58
Joint f_{id}	0.68	0.79	0.74	0.79	0.77	0.76	0.79	0.72	0.76
Joint f_s	0.66	0.71	0.66	0.69	0.75	0.64	0.66	0.70	0.69
Joint $f_{\neg e}$	0.73	0.84	0.78	0.81	0.83	0.78	0.81	0.77	0.79
Joint f_e	0.83	0.94	0.86	0.86	0.87	0.82	0.85	0.84	**0.86**

Table 3: Recognition accuracy (AUC) on AffectNet validation set (neut=neutral, happ=happy, surp=surprise, disg=disgust, ang=anger, cont=contempt; AVG denotes average AUC; ATL=AFFNet-TL, RN=ResNet18, VGG=VGG-Face descriptor).

database after 235k iterations also show high quality (shown as VGG-VGG in Fig.3). Importantly, this network also exhibits good generalization to other databases as shown for reconstructions on the test sets of AffectNet and LFW (VGG-AffectNet and VGG-LFW in Fig.3) - databases that will be used for expression and identity attributes below.

5.3. Discriminative performance

Here, we report discriminative results of the resulting embedding representation after disentanglement training on the different datasets. It is important to stress that given the highly-compressed representation of our architecture, we do not expect our results to beat the state-of-the-art in the field that makes use of very deep, specialized networks with hundreds of millions of parameters. In our experiments, we are interested, first, to check whether a specific facial attribute is encoded only into the respective feature vector and not into the others in order to validate the disentangling process and, second, to chart the performance levels our full pipeline achieves for discriminating each of the face attributes using both sequential training (11k iterations) and additional joint training (another 11k iterations).

5.3.1 Pose

Pose errors for the AffectNet validation set for yaw, pitch, and roll were 3.61°, 3.53°, 2.09°. Results for the CelebA test set were 2.66°, 2.95°, 2.05°. These results are on par (or better) than published results on other datasets such as ALFW, for example, but it should be noted that our ”ground truth” labels consist of the outputs of the face detector rather than crowdsourced labels. Overall, pose detection performance is more than sufficient for our applications.

5.3.2 Identity

We next investigate how well identity information is distilled in the identity sub-feature by performing face verification on the popular Labeled-Faces-in-the-Wild (LFW) benchmark. We trained verification on CelebA using the identify-specific components f_{id} of the feature embedding and its remainder $f_{\neg id}$. In addition, we evaluate for both methods whether joint training of the full pipeline increases performance. Results in Tab.2 show a large performance increase when focusing on the identity sub-vector. In addition, joint training helps to further fine-tune the discriminative information, resulting in 85.4% overall verification performance. After training on the much larger VGGFace2 dataset, results improve further to 95.8% with decreased accuracy in the nonpertinent sub-vectors.

5.3.3 Facial Expressions

We train a small classification network consisting of three 200d fully connected layers. We add dropout (p=0.2) and ReLU layers after the first and second fully connected layer. We train this network on extracted feature vectors form the AffectNet training set. Since the AffectNet training set is highly imbalanced we employ a cross-entropy loss weighted inversely by the number of image in each emotion category. The model is then evaluated on the AffectNet validation set. We report values for the area under the ROC curve (AUC) in Tab.3 after training for 5 epochs. Tab.3 also lists results for each of the different attribute vectors for sequential and joint training as well as comparisons with state-of-the-art results in the literature [30, 40] and a standard pre-trained ResNet18 architecture [12].

Our results again demonstrate that feature information is mostly concentrated in the expression attribute, as f_e has significantly higher scores compared to the other components. Note, however, that residual information about expression is still present in the identity attribute showing that training may not have completely disentangled the information. Again, joint training is shown to boost the results with our final performance being very close to state-of-the-art on this difficult dataset.

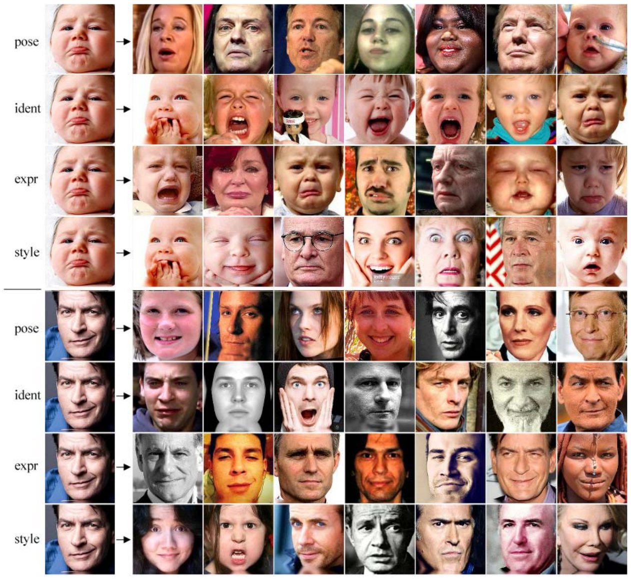

Figure 4: Top-7 images retrieved for two query images on the CelebA dataset.

5.3.4 Face retrieval

We can perform expression-based image retrieval by using nearest neighbor search in the embedding space based on queries that are specific to each attribute (or a combination of attributes). Fig.4 shows results of the Top-7 closest faces on two query images with queries in the CelebA dataset, separated by attribute. Note again, how the system manages to correctly retrieve similar faces according to the specified attribute - queries for identity for the baby face, for example, result in baby faces, and the top results for the actor also contain a correct match.

5.4. Generative capabilities

5.4.1 Face editing

Given the successful disentangling of the face attributes shown above, we next showcase the architecture's capability for face editing. Fig.5 shows a panel of face editing tasks in which information from a target face about a certain face attribute is transferred from a source to a target image. We demonstrate transfer of pose, identity, expression, and style in the first four rows and transfer of all information *except* for pose, identity, expression, and style in rows 5-8.

For the first row, the system performs ”frontalization” given that the source pose is frontal, which works as expected. In the second row, we transfer the identity of the source face into the other targets - for target 3, we see that residual style attributes (lipstick) are still inferred in the transfer, which does, however, still result in a fully realistic face. The expression transfer in the next row is challenging, given the extreme pose and occlusion of the source image, but note how the system correctly transfers a slightly open mouth to the target faces. Style transfer in row 4 also correctly pushes over hairstyle, background, illumination, and skin tone of the source image.

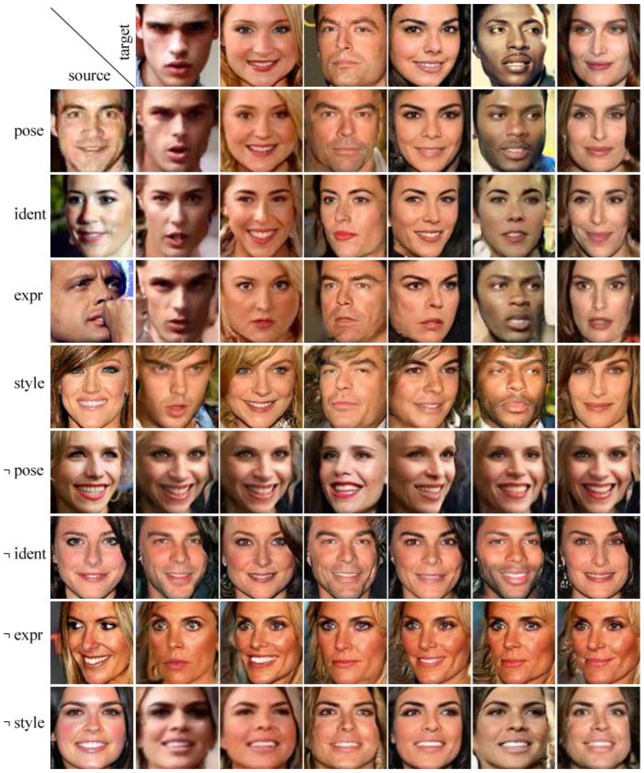

Figure 5: Examples of attribute-specific face editing on random examples from CelebA test set.

In the following four rows, the system should transfer everything except the attribute (which is equivalent to transferring the attribute of the target onto the source). These results demonstrate the consistent transfer of *three* attributes and yield highly consistent synthesized faces.

5.5. Towards explainability

Although our recognition results on facial expressions approach state-of-the-art levels, performance for several classification schemes seems to hover around 0.85AUC. As label quality has been highlighted as a major issue for many cases, including for facial expressions on a different dataset [2], we next explored the confusions that our framework made in this task. Importantly, the use of the generative pipeline allows us to visually inspect each decision as we have access to the autoencoder reconstruction.

Fig.6 shows input faces in the top row together with their AffectNet ground-truth label and reconstructed output faces in the bottom row (color frames encode expression label as well). The first two columns are from correctly-recognized expressions (green dot, happy and surprised ex-

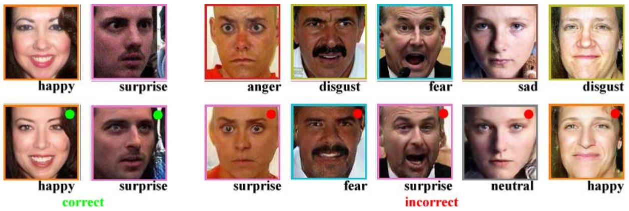

Figure 6: Reconstructed faces and recognized expressions for samples from AffectNet.

pressions) with matching reconstructions. Columns 3-8 are from incorrectly-recognized examples and both the input images and the reconstructed images show that label annotations seem also compatible with the labels predicted from the reconstructed images (see also supplementary material). Hence, similar to [2], annotation quality may be a ceiling for AffectNet at the present moment.

6. Conclusion

In this paper we present an architecture that supports both discriminative and generative tasks by means of a common, low-dimensional face representation. The representation is bootstrapped with an unsupervised learning on a large, unlabeled database and then split into separate components in a subsequent, supervised disentanglement training.

Results show that our system can produce face images with a visual quality at the level of recent GAN approaches for this resolution. The subsequent disentangling process successfully separates different face attributes in the embedding vector as shown by the discrimination experiments with joint post-training of both the encoder and the disentangling providing the best overall results. Not surprisingly, discrimination performance is lower compared to state-of-the-art networks given our low-dimensional embedding and the fact that we rely on a generative framework throughout - nonetheless, performance especially for expression recognition is comparable to state-of-the-art. Our architecture also yields robust, high-quality results for face retrieval and face editing tasks and the generative framework helps to ”debug” database quality and the face representation.

Limitations of the current work include limited reconstruction quality in challenging cases (such as extreme poses) and the presence of typical biases in the training databases. Currently, the disentanglement also does not explicitly deal with age and gender, which will be added as the next step. Additional improvements are necessary to further increase the discriminative performance [34]. Finally, it will be interesting to extend our architecture to other application domains, such as scene or object analysis [29].

References

[1] T. Baltrusaitis, A. Zadeh, Y. C. Lim, and L. P. Morency. OpenFace 2.0: Facial behavior analysis toolkit. *Face and Gesture Recognition, FG 2018*, pages 59–66, 2018. 4, 6

[2] E. Barsoum, C. Zhang, C. C. Ferrer, and Z. Zhang. Training Deep Networks for Facial Expression Recognition with Crowd-Sourced Label Distribution. 2016. 8, 9

[3] D. Bau, J. Zhu, H. Strobelt, Z. Bolei, J. Tenenbaum, W. Freeman, and A. Torralba. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. *arXiv:1811.10597*, 2018. 1

[4] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294, 1988. 2

[5] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018. 6

[6] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *CVPR 2018*, 2018. 5

[7] J. Deng, J. Guo, X. Niannan, and S. Zafeiriou. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *CVPR 2019*, 2019. 1, 2

[8] B. Gecer, B. Bhattarai, J. Kittler, and T. K. Kim. Semi-supervised Adversarial Learning to Generate Photorealistic Face Images of New Identities from 3D Morphable Model. In *LNCS*, volume 11215, pages 230–248, 2018. 2

[9] I. Goodfellow, J. Pouget-Abadie, and M. Mirza. Generative Adversarial Networks. *arXiv:1406.2661v1*, pages 1–9, 2014. 2, 3

[10] G. Gu, S. T. Kim, K. Kim, W. J. Baddar, and Y. M. Ro. Differential Generative Adversarial Networks: Synthesizing Non-linear Facial Variations with Limited Number of Training Data. 2017. 2

[11] H. Han, A. K. Jain, F. Wang, S. Shan, and X. Chen. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE transactions on pattern analysis and machine intelligence*, 40(11):2597–2609, 2018. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Arxiv.Org*, 7(3):171–180, 2015. 5, 7

[13] A. Heljakka, A. Solin, and J. Kannala. Pioneer Networks: Progressively Growing Generative Autoencoder. 2018. 2

[14] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS 2017*, pages 6626–6637, 2017. 6

[15] R. S. Hinton, Geoffrey E; Zemel. Autoencoders, Minimum Description Length and Helmholtz free Energy. *NIPS 1994*, 6:3—-10, 1994. 2

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR 2017*, pages 4700–4708, 2017. 1

[17] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces*

in'Real-Life'Images: detection, alignment, and recognition, 2008. 6

[18] R. Huang, S. Zhang, T. Li, and R. He. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. In *ICCV 2017*, volume 2017-Octob, pages 2458–2467, 2017. 2

[19] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *arXiv 1710.10196*, pages 1–26, 2017. 2

[20] T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv 1812.04948*, 2018. 1

[21] K. Kaulard, D. W. Cunningham, H. H. Bülthoff, and C. Wallraven. The MPI facial expression databasea validated database of emotional and conversational facial expressions. *PloS one*, 7(3):e32321, 2012. 5

[22] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv 1412.6980*, pages 1–15, 2014. 6

[23] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *arXiv 1312.6114*, (Ml):1–14, 2013. 2

[24] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking Clever Hans Predictors and Assessing What Machines Really Learn. *Nature Communications*, 10:1096, 2019. 1

[25] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CVPR 2017*, pages 105–114, 2017. 2

[26] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial Autoencoders. *arXiv 1511.05644*, pages 1–10, 2015. 2

[27] A. Mollahosseini, B. Hasani, and M. H. Mahoor. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, 2017. 4, 5, 6

[28] A. Nagraniy, J. S. Chungy, and A. Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *INTERSPEECH 2017*, pages 2616–2620, 2017. 4, 6

[29] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. In *CVPR 2019*, 2019. 1, 9

[30] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. pages 41.1–41.12, 2015. 7

[31] A. Radford. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. 2015. 6

[32] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, 2019. 2

[33] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed. Variational Approaches for Auto-Encoding Generative Adversarial Networks. *arXiv 1706.04987*, 2017. 2

[34] J. Shao, L. Sheng, J. Yan, and X. Wang. Exploring Disentangled Feature Representation Beyond Face Identification. 2018. 2, 9

[35] V. A. Sindagi and V. M. Patel. {DAFE-FD:} Density Aware Feature Enrichment for Face Detection. *Proc. of WACV2019*, 2019. 2

[36] L. Song, Z. Lu, R. He, Z. Sun, and T. Tan. Geometry Guided Adversarial Facial Expression Synthesis. *arXiv 1712.03474*, 2017. 2

[37] L. Tran, X. Yin, and X. Liu. Disentangled Representation Learning GAN for Pose-Invariant Face Recognition. In *CVPR 2017*, 2017. 2

[38] D. Ulyanov, A. Vedaldi, and V. Lempitsky. It Takes (Only) Two : Adversarial Generator-Encoder Networks. *AAAI 2016*, pages 1250–1257, 2016. 2

[39] T. Valentine. *Cognitive and computational aspects of face recognition: Explorations in face space*. Routledge, 2017. 1

[40] R. Vemulapalli and A. Agarwala. A Compact Embedding for Facial Expression Similarity. *arXiv 1811.11283*, 2018. 2, 7

[41] P. Vuilleumier, J. L. Armony, J. Driver, and R. J. Dolan. Distinct spatial frequency sensitivities for processing faces and emotional expressions. *Nature neuroscience*, 6(6):624, 2003. 3

[42] T.-c. Wang, M.-Y. Liu, J.-y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro. Video-to-Video Synthesis. *arXiv 1808.06601*, pages 1–14, 2018. 2

[43] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *ICCV 2015*, pages 370–378, 2015. 2

[44] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV 2014*, pages 818–833. Springer, 2014. 1

[45] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *CVPR 2017*, pages 4352–4360, 2017. 2

[46] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *ICCV 2017*, pages 2242–2251, 2017. 2