Neural Computing and Applications
https://doi.org/10.1007/s00521-020-04976-7

ORIGINAL ARTICLE

Autoencoder-based image processing framework for object appearance modifications

Krzysztof Ślot[1] · Paweł Kapusta[1] · Jacek Kucharski[1]

Received: 15 June 2019 / Accepted: 25 April 2020

Abstract

The presented paper introduces a novel method for enabling appearance modifications for complex image objects. Qualitative visual object properties, quantified using appropriately derived visual attribute descriptors, are subject to alterations. We adopt a basic convolutional autoencoder as a framework for the proposed attribute modification algorithm, which is composed of the following three steps. The algorithm begins with the extraction of attribute-related information from autoencoder's latent representation of an input image, by means of supervised principal component analysis. Next, appearance alteration is performed in the derived feature space (referred to as 'attribute-space'), based on appropriately identified mappings between quantitative descriptors of image attributes and attribute-space features. Finally, modified attribute vectors are transformed back to latent representation, and output image is reconstructed in the decoding part of an autoencoder. The method has been evaluated using two datasets: images of simple objects—digits from MNIST handwritten-digit dataset and images of complex objects—faces from CelebA dataset. In the former case, two qualitative visual attributes of digit images have been selected for modifications: slant and aspect ratio, whereas in the latter case, aspect ratio of face oval was subject to alterations. Evaluation results prove, both in qualitative and quantitative terms, that the proposed framework offers a promising tool for visual object editing.

Keywords Visual object editing · Convolutional autoencoders · Supervised principal component analysis · Machine learning

1 Introduction

The advent of deep neural networks and deep learning methodologies was a breakthrough in artificial intelligence, enabling its applications in a wide variety of engineering tasks. One of the domains that benefited the most from recent developments is the field of generative models. Deep learning delivered tools that offer generation of realistic visual image objects [4, 10, 30], produce high-quality speech [20, 22, 28], provide language translation [29], summarize contents of documents [23] or generate image descriptions [14, 31].

The largest volume of research on generative models is concerned with visual object synthesis. There are several reasons for that, such as abundance of experimental material, relative ease of evaluation methodology design and high demand for methods offering realistic image generation. Application areas for these methods are very broad and range from entertainment industry and education, which await high-quality virtual and augmented reality systems, through development of advanced visual human-computer interfaces to forensics and privacy protection.

To meet challenging requirements, posed by high human perception sensitivity to even tiny appearance distortions, several deep-learning image generation concepts and architectures were elaborated. The two most important novel ideas in the field are autoencoders (AE), and their extension—variational autoencoders (VAE) [16], as well

✉ Paweł Kapusta
pawel.kapusta@p.lodz.pl

Krzysztof Ślot
kslot@p.lodz.pl

Jacek Kucharski
jkuchars@kis.p.lodz.pl

[1] Institute of Applied Computer Science, Lodz University of Technology, Lodz, Poland

Published online: 28 May 2020

as generative adversarial networks (GAN) [8]. In both cases, target visual objects are decoded from some core, latent representations via a neural generative system. Probability densities of these representations, as well as parameters of decoding algorithms, need to be learned from examples.

Having developed a framework for generating visual objects, intense research on gaining more control over generation outcome properties was launched. Several methods that enable editing object's appearance, which target various high-level visual attributes, have been proposed so far. For example, remarkable results of research on generating facial images with specific facial expressions [2], on modeling of aging [1], gender-switching [33] or adding/removing extra contents, such as eyeglasses or makeup [26], were reported.

The presented paper introduces a novel approach to image object visual appearance editing. The proposed concept offers functional control over expression intensity of selected qualitative, yet measurable, object's appearance attributes. For example, qualitative notion of face slimness can be coarsely quantified using a ratio of axes of a face-oval approximating ellipse. To enable editing of such attributes, we propose to establish (via learning) and to exploit functional relationship between visual object's appearance and attribute's estimate. It is important to note that one cannot get plausible appearance modification effects via simple geometric visual object transformations, e.g., image region scaling. Therefore, sufficiently complex data processing methods, which discover relations between visual contents and its pixel-level realization, need to be developed.

The proposed appearance modification scenario is to first extract information related to image object's target attributes, and then, to purposefully modify this information and finally, to reassemble a new, modified image with altered contents. A natural environment for performing these operations is offered by a latent space of a convolutional autoencoder [19] (CAE), so this paradigm has been adopted as a computational framework for the method implementation. Convolutional autoencoders, trained to accurately reproduce image objects that are to undergo modifications, develop, through nonlinear transformations of raw input data, latent, semantic representations of image contents.

The proposed method comprises the following three components. The first one is extraction of information on target appearance attribute from CAE's latent representation. To extract this information, we propose to apply supervised principal component analysis [3] (abbreviated as SPCA), which attempts to appropriately combine pieces of attribute-related information, scattered among latent vector components, into features that strongly correlate with target attributes. The second element of the proposed scheme is an attribute-modification module. The SPCA-derived feature space (referred to as *attribute-space*) forms a domain that is expected to enable convenient formulation of functional, monotonous mappings between its component features and considered appearance attributes. The derived mappings are subsequently used for introducing appearance modifications: A required change in appearance, quantified by attribute-descriptor change, gets transformed onto a modified version of attribute-space vectors. The last element of the proposed data processing pipeline is to assemble edited versions of latent vectors, which is performed by the inverse SPCA.

The outcome of the presented appearance modification scheme is finally decoded onto an output image in the decoding part of the CAE. Two training procedures are executed to build the proposed appearance editing system. First, CAE is trained on unlabeled examples of visual objects of interest, and then, SPCA transformation parameters and attribute-space feature mappings are learned using an additional set of labeled training examples.

The main contribution of the paper is the formulation of the novel visual object editing framework. We propose a data processing scheme that is different from the existing paradigms: Contents related to a specific appearance attribute are extracted from latent space through transformation, which decorrelates it from information on other high-level visual properties, so that it can be selectively manipulated without affecting the remaining components. This contrasts with other approaches, which apply direct transformations to latent space vectors, and therefore, require additional mechanisms for disentangling existing between-attribute dependencies. Another key distinction between the proposed concept and existing visual object attribute modification schemes is a functional relation that is established between appearance attribute and its produced, low-level representation. This enables introducing continuous changes (rather than discrete ones) to appearance in a simple manner.

The proposed algorithm has been evaluated using both simple and complex visual objects. In the former case, handwritten digits from the MNIST dataset [18] were selected as modification targets. Two different digit's appearance attributes—slant and object's proportions, were considered for editing. In the latter case, processing targets were face images from CelebA dataset [34], and face slimness was chosen as an attribute to be modified. In both cases, evaluation results show that appearance modifications induced by the proposed attribute control mechanism match qualitative expectations, and efficacy of the method is supported by proposed quantitative indicators of its performance.

The structure of the paper is the following. We begin with brief summary of related work in Sect. 2. We provide detailed explanation of the proposed framework in Sect. 3, and we present experimental evaluation results of the proposed algorithm in Sect. 4.

2 Related work

Although the two aforementioned deep generative models: autoencoders and generative adversarial networks, have been introduced only recently, an impressive amount of successful research on complex visual object generation has been presented thus far [5–7, 9, 11, 25, 27, 32, 35]. Basic ability to generate realistically looking visual objects of a specified category has quickly expanded to provide more detailed control over objects' visual properties. A notable development was the introduction of a conditional GAN (cGAN) concept [21], where a mechanism for conditioning generation outcome on additional information has been proposed. The conditioning was implemented as an additional component of an objective criterion used both in discriminator and generator training. Although the concept was originally aimed only at enabling selection of specific modes of learned distributions (e.g., a particular digit) or for binding external modalities (e.g., textual descriptions), it quickly evolved to become a general framework for inserting information on target image attributes. For example, [24] proposes to extend cGAN model with an encoder that retrieves attribute-related information from images, so that it can be later modified and assembled with latent representation, yielding required appearance modifications. Remarkable results of application of this scheme in altering visual attributes include hair color, facial expressions or addition of eyeglasses. A similar concept is proposed in [2], where a 'connection network' provides a trainable mapping between attributes and the corresponding image space. It should be noted that the proposed framework also enables introducing continuous modifications of attribute-expression intensity.

Two important developments in the field of cGANs include IcGAN [24] and Fader networks [17], which attempt to enforce attribute-independent latent representation during training and then to supplement it with attribute-specific input, for the purpose of target visual object synthesis. However, the most promising cGAN-based architecture, with remarkable attribute-editing capabilities, is AttGAN [12]. The AttGAN algorithm offers a means for generating high-quality images with attributes chosen from a discrete candidate set. This is accomplished by introducing computational architecture aimed at attaining three different objectives: attribute-independent reproduction of input objects, learning of appropriate attribute re-mappings and generation of visually plausible outcome. In addition to discrete attribute changes, AttGAN also enables continuous control over attribute-expression intensity.

The most notable approach to object generation and editing within a framework of variational autoencoders, elaborated in [33], uses separate encoder-decoder pairs for handling appropriately defined foreground and background image components. The former component is learned by the autoencoder using a criterion that involves an additional attribute information, where attributes are specified as categorical, natural language variables.

A common denominator for all of the proposed concepts for visual object editing is a parallel mechanism for latent and attribute-specific information extraction. In contrast to this framework, we propose to use a 'serial' data processing scheme, where attribute-related content is sought in latent representation. Moreover, all approaches proposed so far attempt to impose direct alterations to image latent representations, whereas the proposed method modifies latent representation implicitly, via additional transformation layer. Finally, unlike most existing GAN-based or autoencoder-based methods, the proposed approach is aimed at introducing appearance modification to a specific input object, so that it can be considered as a tool for attribute-based visual object editing.

3 The method

An objective of the research was to develop a method for enabling control over qualitative appearance of complex visual objects. We assumed that appearance could be quantified using a set of high-level ordinal attributes (e.g., object's slant or aspect ratio), and we adopted a pre-trained convolutional autoencoder as an image generation algorithm.

CAE performs two data processing steps. Firstly, it maps input samples onto some latent, d-dimensional space $\mathcal{Z}$, so that a given input object $\mathbf{x}$ gets transformed to its corresponding latent vector:

$$\mathbf{z} = \Phi(\mathbf{x}) \tag{1}$$

In the second step, it attempts to reconstruct an original image from the produced latent representation:

$$\mathbf{x} \approx \Phi^{*}(\mathbf{z}) \tag{2}$$

where Φ^{*} is expected to closely approximate Φ^{-1}.

An objective of appearance modification algorithm to be developed can be formally stated as a search for an appropriate transformation Ψ that modifies contents of a latent vector $\mathbf{z}^{i}$, which encodes an original input image $\mathbf{x}^{i}$ and produces a latent vector $\mathbf{z}^{o}$ that is subsequently decoded onto a modified output image $\mathbf{x}^o$ with purposely altered appearance. The expected transformation can be expressed as:

$$\mathbf{x}^o = \Phi^*(\mathbf{z}^o) = \Phi^*\big(\Psi(\mathbf{z}^i)\big) = \Phi^*\big(\Psi\big(\Phi(\mathbf{x}^i)\big)\big) \qquad (3)$$

The proposed method for latent vector modifications, which implements the mapping $\mathbf{z}^o = \Psi(\mathbf{z}^i)$ and appropriately alters object properties according to (3), is explained in the following subsection.

3.1 Attribute modification procedure

Space $\mathcal{Z}$ of latent variables $\mathbf{z}$ provides some unknown encodings for all visual properties of objects generated by an autoencoder, including information related to attributes one would like to alter (which are, henceforth, referred to as *target attributes*). However, this information is most likely distributed among many (possibly, all) components of a latent vector. In order to enable selective manipulations that alter only target attributes and leave the remaining ones intact, a method for uncoupling these components should be elaborated.

A possible way for extracting information related to target attributes is to apply suitable, invertible mapping that transforms the latent space to a new one, where relevant information is concentrated in just a few components. This new space is referred to as an attribute-space, as its principal directions are expected to correlate well with target attributes. Since only a few dominant components of attribute-space vectors would become subject to modifications, we expect that a negative impact on other object's visual properties would be minimized. As the main objective of the transformation to be applied is to extract features that provide strong correlations between raw data (latent vectors) and labels (appearance attribute descriptors), supervised principal component analysis (SPCA) becomes a natural candidate to do the task.

The proposed computational architecture that implements the scheme defined by (3) is schematically depicted in Fig. 1. Let's assume that we are given a pre-trained convolutional autoencoder (i.e., the mapping: $\mathbf{z} = \boldsymbol{\Phi}(\mathbf{x})$ has been determined) and that we are given an additional set of n labeled images. It implies that we can build a matrix $\mathbf{Z}$ of d-dimensional latent vectors $\mathbf{z}$, that encode these input images:

$$\mathbf{Z} = \left[\mathbf{z}^1 \ldots \mathbf{z}^n\right]_{d\times n} = \left[\boldsymbol{\Phi}(\mathbf{x}^1)\ldots\boldsymbol{\Phi}(\mathbf{x}^n)\right] \qquad (4)$$

Each latent vector $\mathbf{z}^k$ is labeled with a q-element vector $\mathbf{a}^k = [a_1 \ldots a_q]^T$ that quantifies appearance attributes of the corresponding input image $\mathbf{x}^k$.

The first functional block of the proposed architecture implements derivation of the attribute space by means of SPCA. SPCA transformation matrix: $\mathbf{S} = [\mathbf{s}^1 \ldots \mathbf{s}^d]_{d\times d}$ comprises d-eigenvectors of the mapping that maximizes linear dependence between latent-space vectors and their labels, expressed using cross-covariance matrix $\mathbf{C}_{\mathbf{z},\mathbf{a}}$ defined as:

$$\mathbf{C}_{\mathbf{z},\mathbf{a}} = E\big((\mathbf{z} - \boldsymbol{\mu}_z)(\mathbf{a} - \boldsymbol{\mu}_a)^T\big) \qquad (5)$$

where $\boldsymbol{\mu}_z, \boldsymbol{\mu}_a$ are latent vector and label vector means. A criterion underlying SPCA basis derivation has the form:

$$J(\mathbf{s}) = \max_{\mathbf{s}}\ \mathbf{s}^T tr(\mathbf{C}_{\mathbf{z},\mathbf{a}}\mathbf{C}^T_{\mathbf{z},\mathbf{a}})\mathbf{s}, \quad ||\mathbf{s}^T\mathbf{s}|| = 1 \qquad (6)$$

where $tr(.)$ denotes a trace.

Transformation of a latent vector $\mathbf{z}$ from a latent space to an attribute space can be expressed as:

$$\xi = \mathbf{S}^T\mathbf{z} \qquad (7)$$

It is expected that after this transformation, only a few, p-leading components ($p \ll d$) of the basis $\mathbf{S}$ contain information related to target attributes, so only p-element subsets of attribute-space vectors $\xi_{\mathcal{P}}$ will be subject to modifications:

$$\xi_{\mathcal{P}} = (\mathbf{S}_{\mathcal{P}})^T\mathbf{z}, \quad (\mathbf{S}_{\mathcal{P}})^T = \begin{bmatrix} (\mathbf{s}^1)^T \\ \cdots \\ (\mathbf{s}^p)^T \end{bmatrix}_{p\times d} \qquad (8)$$

Mechanisms for modifying visual object's appearance are implemented in the 'Attribute change' block of the proposed architecture (Fig. 1). An idea is to derive, based on labeled training examples, functional mappings between a given appearance-attribute descriptor a_k ($a_k \in \{a_1 \ldots a_q\}$) and the contents of sub-vectors $\xi_{\mathcal{P}}$, for all considered visual appearance attributes:

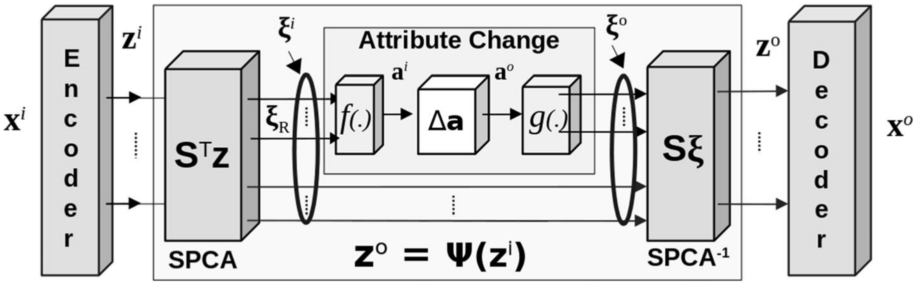

Fig. 1 Autoencoder architecture with the proposed attribute modification module

$$a_k = f_k(\xi_{\mathcal{P}}),\ \ \xi_{\mathcal{P}} = g_k(a_k) \tag{9}$$

The mapping $f_k(.)$ will be referred to as *attribute-scoring* function, whereas the mapping $g_k(.)$, which inverts the transformation $f_k(.)$, will be referred to as *attribute-encoding* function for a k-th appearance attribute. The purpose of the attribute-scoring is to evaluate a current state of some considered, k-th appearance property encoded in the attribute vector, whereas the latter function encodes appropriately updated states.

Appearance modification scenario is thus a sequence of three operations. First, a portion $\xi^i_{\mathcal{P}}$ of a current attribute vector ξ^i obtained for input image $\mathbf{x}^i$ is evaluated using scoring-functions for all considered attributes:

$$\mathbf{a}^i = [f_1(\xi^i_{\mathcal{P}})..f_q(\xi^i_{\mathcal{P}}] \tag{10}$$

Next, the evaluated appearance attributes get modified to match desired, new values:

$$\mathbf{a}^o = \mathbf{a}^i + \Delta\mathbf{a} \tag{11}$$

Finally, the modified attribute values are transformed by attribute-encoding functions:

$$\xi^o_{\mathcal{P}} = [g_1(\mathbf{a}^o)...g_q(\mathbf{a}^o))] \tag{12}$$

and are used to update the p-leading positions of an original attribute-space vector:

$$\xi^o = [\xi^o_{\mathcal{P}} | \xi^i_{p+1} \cdots \xi^i_d] \tag{13}$$

The last operation of the proposed computational architecture is an inverse SPCA transformation of the attribute space back to the latent space. Since a basis $\mathbf{S}$ is orthonormal, this can be expressed as:

$$\mathbf{z}^o = \mathbf{S}\xi^o \tag{14}$$

Finally, the resulting latent vector $\mathbf{z}^o$ is decoded onto an output image $\mathbf{x}^o$ in the decoding part of Convolutional Autoencoder.

4 Experimental evaluation of the proposed concept

An experimental evaluation of the proposed concept has been done using two datasets. The first one—MNIST handwritten digits dataset, which comprises relatively simple binary objects, has been subject to several thorough analyses that examine the importance of various parameters of the proposed algorithm. The second one—CelebA face dataset, has been used to demonstrate capabilities of the proposed method in handling appearance modifications of very complex visual objects.

An objective of the experiments was to verify whether the proposed concept enables control over object's appearance properties. Two appearance attributes that are relatively easy to quantify: slant and aspect ratio, were considered for handwritten digits. In the case of faces, slimness, represented using aspect ratio of face-oval approximating ellipse, was subject to modifications.

Data processing pipeline that was implemented corresponds to the diagram shown in Fig. 1. First, CAE is trained on unlabeled examples, and then, SPCA and attribute-scoring and attribute-encoding functions are derived using labeled example subset. Processing of input sample $\mathbf{x}^i$ begins with its transformation to a latent space ($\mathbf{z}^i$), followed by a projection of $\mathbf{z}^i$ to the attribute-space (ξ^i). After attribute change, the resulting vector ξ^o is transformed subsequently to $\mathbf{z}^o$ and $\mathbf{x}^o$.

4.1 Autoencoder training scenarios

Different autoencoder architectures and autoencoder training scenarios were tested throughout experiments. Details of the considered convolutional autoencoders are provided in Table 1 for experiments with MNIST digits and in Table 2, for experiments with CelebA faces. CAE for MNIST was trained using the Adadelta optimizer with a learning rate of 1.0 and a decay factor of 0.95. The batch size was set to 128. CelebA autoencoder was trained using Adam optimizer ($\beta 1 = 0.9$, $\beta 2 = 0.999$) with the learning rate of 1×10^{-4} and a batch size of 16.

In the case of digit appearance modifications, two experimental scenarios were adopted. The first one

Table 1 Details of CAEs used in digit appearance modifications

Encoder	Decoder
Conv2D (16, 3, 1), ReLU	Conv2D (2, 3, 1)[1]
	Conv2D (4, 3, 1)[2]
	Conv2D (8, 3, 1)[3]
	ReLU
MaxPooling2D (2, 2)	UpSampling2D (2, 2)
Conv2D (8, 3, 1), ReLU	Conv2D (8, 3, 1), ReLu
MaxPooling2D (2, 2)	UpSampling2D (2, 2)
Conv2D (2, 3, 1)[1]	Conv2D (16, 3, 1), ReLU
Conv2D (4, 3, 1)[2]	
Conv2D (8, 3, 1)[3]	
ReLu	
MaxPooling2D (2, 2)	UpSampling2D (2, 2)

*Conv2D(d,k,s) denotes the 2D convolutional layer with d as dimension, k as kernel size and s as stride

[1,2,3] Denotes the size of the encoded representation: 1—32, 2—64, 3—128

Table 2 Details of CAEs used in face appearance modifications

Encoder	Decoder
Conv2D (256, 6, 1), ReLU	Dense (9152)
GaussianDropout (0.3)	Reshape (target = 13, 11, 64)
Conv2D (256, 6, 1), ReLU	DeConv2D (128, 2, 1), ReLU
GaussianDropout (0.3)	DeConv2D (128, 2, 2), ReLU
MaxPooling2D	DeConv2D (64, 3, 1), ReLU
Conv2D (128, 5, 1), ReLU	DeConv2D (64, 3, 2), ReLU
GaussianDropout (0.3)	DeConv2D (64, 4, 1), ReLU
Conv2D (128, 5, 1), ReLU	DeConv2D (64, 4, 2), ReLU
GaussianDropout (0.3)	DeConv2D (64, 3, 1), ReLU
MaxPooling2D	DeConv2D (64, 3, 2), ReLU
Conv2D (128, 4, 1), ReLU	DeConv2D (64, 4, 1), ReLU
GaussianDropout (0.3)	DeConv2D (64, 2, 1), ReLU
Conv2D (128, 4, 1), ReLU	DeConv2D (3, 3, 1), ReLU
GaussianDropout (0.3)	
MaxPooling2D	
Conv2D (128, 3, 1), ReLU	
GaussianDropout (0.3)	
Conv2D (128, 3, 1), ReLU	
GaussianDropout (0.3)	
MaxPooling2D	
Conv2D (128, 2, 1), ReLU	
GaussianDropout (0.3)	
Conv2D (128, 2, 1), ReLU	
GaussianDropout (0.3)	
Flatten (9152)	
Dense [1,2,3,4]	

*Conv2D(d,k,s) and DeConv2D(d,k,s) denote the 2D convolutional layer and 2D transposed convolutional layer with d as dimension, k as kernel size and s as stride

[1,2,3,4] Denotes the size of the encoded representation: 1—1024, 2—2048, 3—4096, 4—8192

assumes that autoencoder is trained on digits from all ten classes (*digit-independent autoencoder*), whereas the second one assumes that we derive separate autoencoders for each digit (*digit-specific autoencoders*). In the former scenario, autoencoder was trained on sixty thousand examples. From the remaining portion of the MNIST dataset, a subset of 3500 digits (350 elements per class) was selected, to form a basis for deriving the proposed attribute-modification module. All elements of this set were labeled with attribute descriptors, and 80% of them were used to compute SPCA transformation matrix, whereas the remaining 20% were used for method's testing. In the second training scenario, 6000 examples per class were used to derive ten different autoencoders, specializing in handling digits from a particular class. For each autoencoder, the SPCA matrix was derived based on manually labeled 350 samples of the corresponding class. For all considered scenarios, test and training sets were disjoint: a digit that was subject to attribute modifications was neither previously used in autoencoder training nor in SPCA/attribute-scoring function derivation.

In case of face appearance modification experiments, CAE was trained on all 200,000 samples from the CelebA database. A subset of 32,418 images was labeled with the proposed face slimness descriptor and used in training and testing of the proposed method.

4.2 Appearance attribute descriptors

To derive appearance descriptors for the adopted digit properties: its slant and its proportions, we treat binary images as distributions of foreground pixels in a 2D space. Covariance matrices calculated for these distributions form a basis for derivation of quantitative descriptors for both attributes. Digit's slant is assessed as an angle between the image coordinate system vertical axis and the dominant eigenvector of the covariance matrix. Digit's aspect ratio (denoted by AR) is evaluated as a ratio of covariance matrix eigenvalues. As a result, each element $\mathbf{x}^k$ of training and test sets, processed using the proposed attribute-modification module, is assigned a set of two labels: $\boldsymbol{a}^k = [a_S^k, a_{AR}^k]$.

Face slimness is evaluated using an aspect ratio (AR) of face-oval approximating ellipse axes (i.e., a *b*/*a* ratio, see Fig. 2). Elliptic approximations of face-ovals were determined based on subsets of facial landmarks extracted for training images using the detector available at dlib library [15].

4.3 Appearance representation in latent space

There were two main reasons for discarding a latent space to be the right domain for high-level attribute modifications. We expected attribute representations to be distributed among different components of a latent space. This would imply a necessity of deriving high-dimensional attribute-scoring functions, which could pose a difficult problem. Moreover, we hypothesized that such attribute-

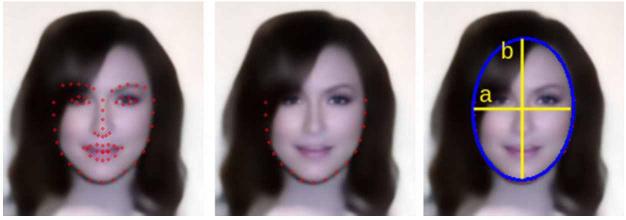

Fig. 2 Face aspect ratio evaluation: face image with detected facial landmarks (left), landmarks considered in face shape estimation (middle) and face-approximating ellipse (right)

scoring functions could be not strictly monotonous, thus non-invertible, so that attribute control would become impossible.

To verify these conjectures, we computed Spearman's correlation coefficients between target attributes and latent vectors, for all considered datasets. Sample results presented for MNIST Digits dataset in Fig. 3 show that majority of latent vector components are correlated with the targets (all results are statistically significant at $p\,\text{value} = 0.01$). Furthermore, a strength of correlations is moderate, which means that no strong monotonous relations exist between a target attribute and particular latent features.

4.4 Appearance representation in attribute-space

We found that only the leading components of SPCA-derived attribute space contained meaningful information on selected appearance attributes. It turns out that if latent vectors were labeled using a single attribute only (either slant or AR for digits or AR for faces), there was just one dominant eigenvalue, whereas for two-element label vectors ($\mathbf{a} = [a_S, a_{AR}]$), there were two dominant components. This observation holds both for digit-specific and digit-independent CAEs.

It follows that attribute-scoring functions in the attribute space are in fact defined over either 1D space or 2D space. To assess chances for getting monotonous attribute-control mappings, we again evaluated Spearman correlation coefficients between the considered attributes and the dominant components of attribute spaces, derived for all considered scenarios.

Sample results, presented for MNIST Digit dataset in Tables 3 and 4 show that SPCA-derived representations form a promising basis for regression, especially for cases, where Spearman correlations reach 100%. However, no clear premises as to autoencoder structure choice nor training scenario selection could be drawn from these experiments.

Table 3 Absolute values of Spearman correlations between the considered attributes and dominant components of attribute spaces for ten 32-line digit-specific AEs

	'1'	'2'	'3'	'4'	'5'
S	1.00	0.87	0.95	0.93	0.90
AR	0.83	0.70	0.81	0.96	0.92
	'6'	'7'	'8'	'9'	'0'
S	0.93	0.90	0.96	0.95	0.99
AR	0.90	0.82	0.82	0.90	0.81

Table 4 Spearman correlations between the considered attributes and dominant components of semantic spaces for digit-independent autoencoders

	32-line AE	64-line AE	128-line AE
S	0.93	0.94	0.90
AR	0.84	0.78	0.81

One-dimensional plots of appearance attribute descriptors versus the dominant attribute space features are shown in Fig. 4, for digits, and in Fig. 5, for faces. Despite variance that in some cases is significant (especially for face images), one can observe a consistent, monotonous relation between the variables, which can be reasonably approximated using different parametric functions. Three different candidates for solving the presented regression problem were considered in case of digit appearance modifications (second- and third-order polynomials as well as multiple-layer, feedforward neural network), whereas only third-

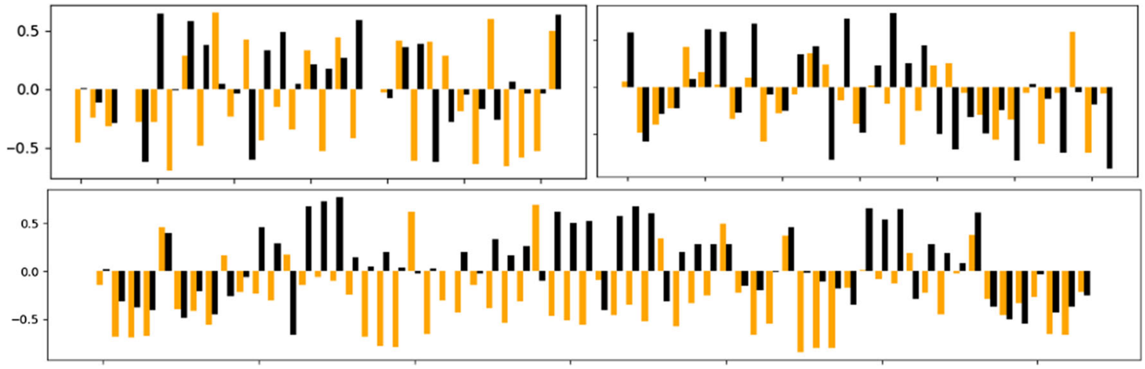

Fig. 3 Spearman correlations between latent vectors and digits' slant (black), and latent vectors and digits' aspect ratios (orange) for: 32-line digit-specific AE trained on a digit 'eight' (top left), 32-line digit independent AE (top right) and 64-line digit-specific AE, also trained on a digit 'eight' (bottom) (colour figure online)

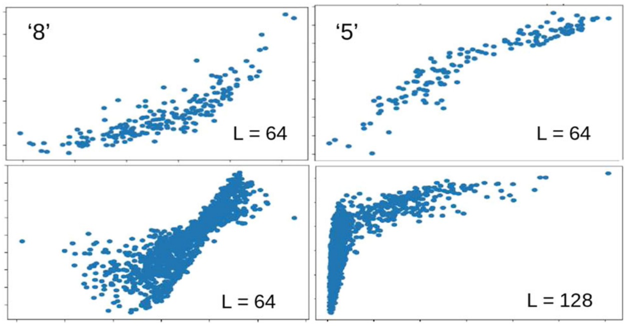

Fig. 4 Scatter plots showing dominant, SPCA-derived feature values (vertical axis) as a function of slant (left column) and aspect ratio (right column) for images of digits, where L is latent representation size. Top row shows results for digit-specific CAEs (trained for digits '8' and '5'), bottom row—for digit-independent CAEs

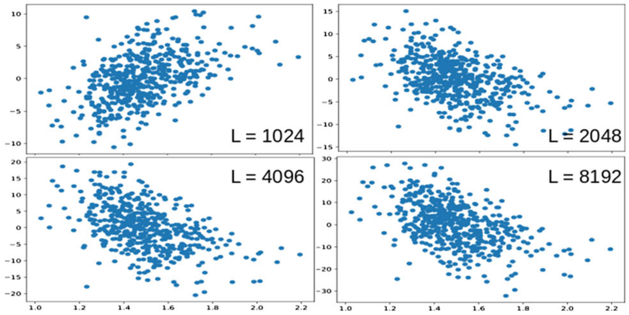

Fig. 5 Scatter plots showing dominant, SPCA-derived feature values (vertical axis) as a function of face aspect ratio (horizontal) for four different sizes of latent representation L

order polynomial was considered for fitting data in case of face appearance modifications.

4.5 Digit appearance modification results

Experimental evaluation of the proposed concept in case of the MNIST dataset involved several scenarios, differing by:

- CAEs architectures: latent spaces of size 32, 64 and 128 lines were considered
- Targeted objects: digit-dependent and digit-independent CAEs were used
- Attribute mapping approximations: three regressors: second- and third-order polynomials and a neural network were considered,

Given the trained architecture, the previously explained test sample processing pipeline, involving its modification in the attribute-space, was executed. The assumed appearance attribute modification range was $\pm 15^\circ$ from the initial angle for slant and $\pm 20\%$ from the initial value for aspect ratio. These initial values were evaluated for a digit generated at the CAE's output (not for the inputted image, as CAE can introduce appearance alterations).

Given target (t_i^k) and actual (y_i^k) values of an attribute 'i' for processing of a k-th input sample, a digit transformation accuracy was assessed using the mean relative attribute modification error, i.e.:

$$e_r = \frac{1}{n}\sum_{k=1}^{n}\frac{|t_i^k - y_i^k|}{t_i^k} \tag{15}$$

where n is a number of valid output digits. To assess validity of generated digit images, we performed classification of AE outcome, using a deep convolutional neural network classifier trained on MNIST dataset. For the considered range of attribute modifications, 19% of digits generated by 32-line AE were incorrectly recognized (output digits were overly distorted) and only approximately 5% of digits generated by 64- and 128-line ones were rejected.

A summary of quantitative evaluation results for the considered test scenarios are shown in Fig. 6 (for slant) and Fig. 7 (for aspect ratio). As it can be seen from Fig. 6, neural regressor was very unstable, often producing very poor results, so in the remaining analyses we used only polynomial attribute-scoring functions.

Mean attribute modification errors for various autoencoders show that for all considered cases, digit/independent AEs were performing consistently better than digit-specific ones. The gain was an attribute/independent function of a latent-space dimension and ranged from 4% for 32-line AE to 7% for 128-line AE. A possible reason for this phenomenon could be a larger amount of training samples that were available for training digit-independent AEs.

The best digit-wise attribute manipulation accuracy is summarized in Table 5. One can observe that in 50% of cases, the most accurate results were produced by the simplest, 32-line AE. Slant modification relative errors vary from 18% for the digit 'nine' to 54% for 'five.' Similarly, for the aspect ratio, the best result—16% is reported for the digit 'four,' whereas the worst one—63% is for the digit 'three.' We found statistically insignificant differences between quadratic and cubic attribute-scoring functions.

The last quantitative analysis that has been performed was evaluation of attribute cross-modifications, i.e., changes in slant induced by aspect ratio modifications and changes in aspect ratio caused by slant modifications. The results, presented in Table 6, suggest that as aspect ratio is more sensitive to alterations made for slant, information on aspect ratio may be more diffused over attribute-space components than information on slant (as indicated by Spearman correlations provided in Tables 3 and 4).

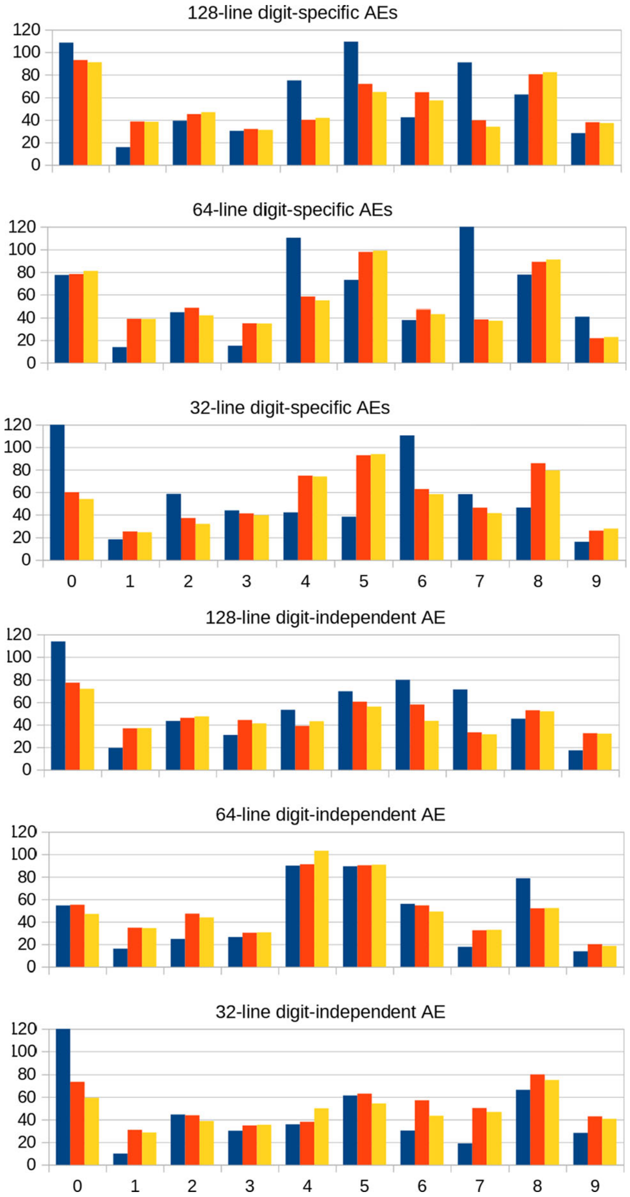

Fig. 6 Slant modification errors for different, digit-specific and digit-independent CAEs and different regression functions (blue bars—neural network, red—second- and yellow—third-order polynomials) (colour figure online)

Sample qualitative results of digit slant and aspect ratio modifications are presented in Fig. 8. The images shown were generated by digit-independent AEs with 32-dimensional latent space and a second-order polynomial used as the attribute-scoring function. For all digits, aspect ratio was altered within a range of ±20% with respect to an initial value and slant—within a range of ±15° . Mean attribute modification errors for the presented renderings were 18% for aspect ratio and 26% for slant.

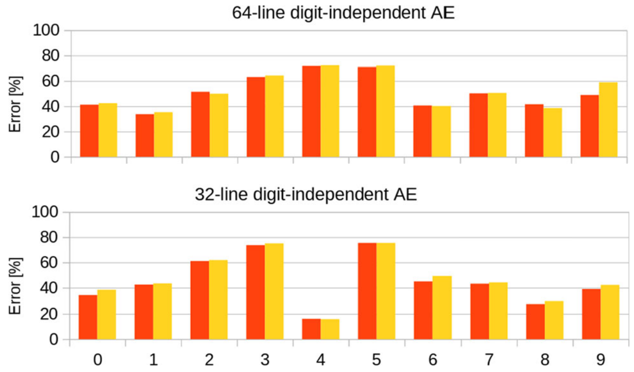

Fig. 7 Aspect ratio modification errors for the best performing AEs (64- and 32-line), all digits and polynomial attribute-scoring functions

Table 5 Digit-wise attribute modification relative errors, in percent (for slant—*S* and for aspect ratio—AR), for best performing CAEs (*L*—latent space dimension)

	0	1	2	3	4	5	6	7	8	9
S	47	28	38	30	38	54	43	31	31	18
L	64	32	32	64	32	32	128	128	32	64
AR	34	33	49	63	16	50	40	44	27	39
L	32	64	64	64	32	128	64	32	32	32

Table 6 Mean sensitivity of digit's slant to its aspect ratio modifications (ΔS, in percent) and mean sensitivity of digit's aspect ratio to modifications of its slant (ΔAR, in percent), for all considered scenarios

	Digit-independent AEs			Digit-specific AEs		
	32	64	128	32	64	128
ΔS	9.85	9.39	7.52	9.31	8.9	7.1
ΔAR	23.28	24.17	19.82	23.08	25.46	21.71

4.6 Face appearance modification results

A procedure used in case of face appearance modification was the same as previously described for digits. This time much more complex CAEs were applied (see Table 2) to match the larger size of input images (208 × 176 pixels, 3 channels) and much richer image contents. Four different cardinalities of CAE's latent image representation were considered: $L = 1024, L = 2048, L = 4096$ and $L = 8192$. As it was previously explained, face slimness was the only appearance attribute subject to modifications, and third-order polynomials were fit to approximate mappings required by the attribute-change module of the algorithm. Four modifications of face-oval ellipse aspect ratio were

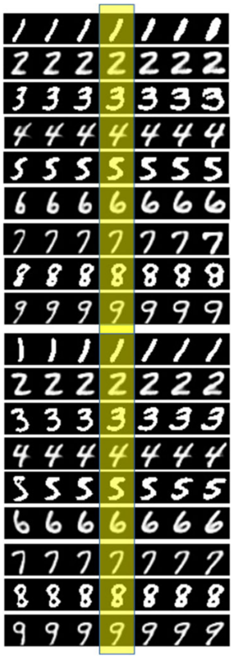

Fig. 8 Sample images generated from highlighted digits by altering: aspect ratio (top) and slant (bottom)

introduced to each test image during the experiments: − 20%, − 10%, + 10% and + 20%.

The first batch of experiments was aimed at assessing the capacity of different considered autoencoders to implement the expected transformations. Quantitative results (see Fig. 9) show, for extreme modifications, that all architectures are capable of introducing plausible transformations. This means that even the most compressed, 1024-element representation of images contains all necessary information on face oval shape.

Sample face slimness editing results, performed using CAE with latent vectors of width $L = 4096$ for different faces, have been presented in Fig. 10. As it can be seen, modifications result in noticeable face appearance changes, yet they do not impair appearance plausibility. (For the considered images, artifacts in the form of implausible shades can be seen for increased aspect-ratios.) Also, one can observe that appearance change is gradual, although it is difficult to subjectively asses, whether the introduced, evenly spaced modifications induce proportional face appearance changes.

Quantitative assessment of the proposed face appearance modification method is provided in Table 7. Accuracy of induced changes is expressed through the mean relative error (as defined by (15)) between target and actual aspect ratios of face-oval approximating ellipses. The experiments were performed for all considered autoencoder architectures. As can be seen, results are comparable, which supports conclusions drawn for subjective evaluation of results presented in Fig. 9. Relatively large errors for extreme modifications of face slimness may partly result from the fact that in several cases, target ratios exceed a range of attribute values present in the training set.

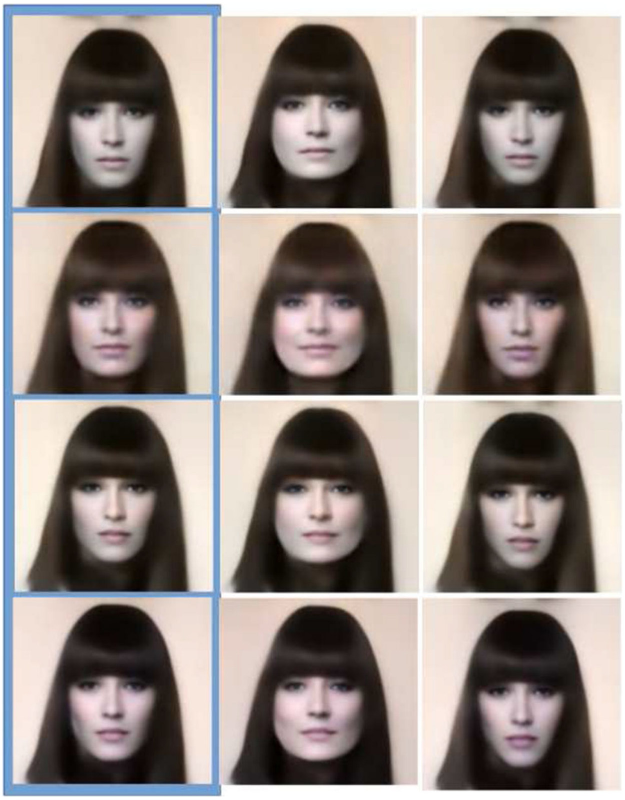

Fig. 9 Face appearance modifications in CAEs with different size of latent representation. From top to bottom: 1024, 2048, 4096 and 8192 lines. Left column (highlighted): original appearance, middle column: aspect ratio reduced by 20%, right column—aspect ratio increased by 20%

To compare the proposed procedure with other existing approaches, an AttGAN implementation provided in [13] was used to induce face-oval alterations. However, for the considered face dataset, we were unable to train AttGAN to enable introducing the expected slimness modifications.

5 Conclusions

The presented method provides a framework for manipulating visual object's appearance, through affecting high-level appearance attributes. The provided evaluation proves that one can ensure coarse, quantitative control over complex visual properties of a specific image object.

The proposed algorithm has been implemented as a part of a convolutional autencoder, as it enables editing of a particular object. However, as autoencoding is a general concept, one can also investigate possibilities of extending the proposed method to control high-level attributes that characterize sequences, e.g., emotional speech.

Fig. 10 Face aspect ratio modifications for two female and two male face images. The first column shows faces with aspect ratios reduced by 20%, the second by 10%, middle column (highlighted) presents original faces and subsequent columns correspond to increased aspect ratios by 10% and 20%, respectively

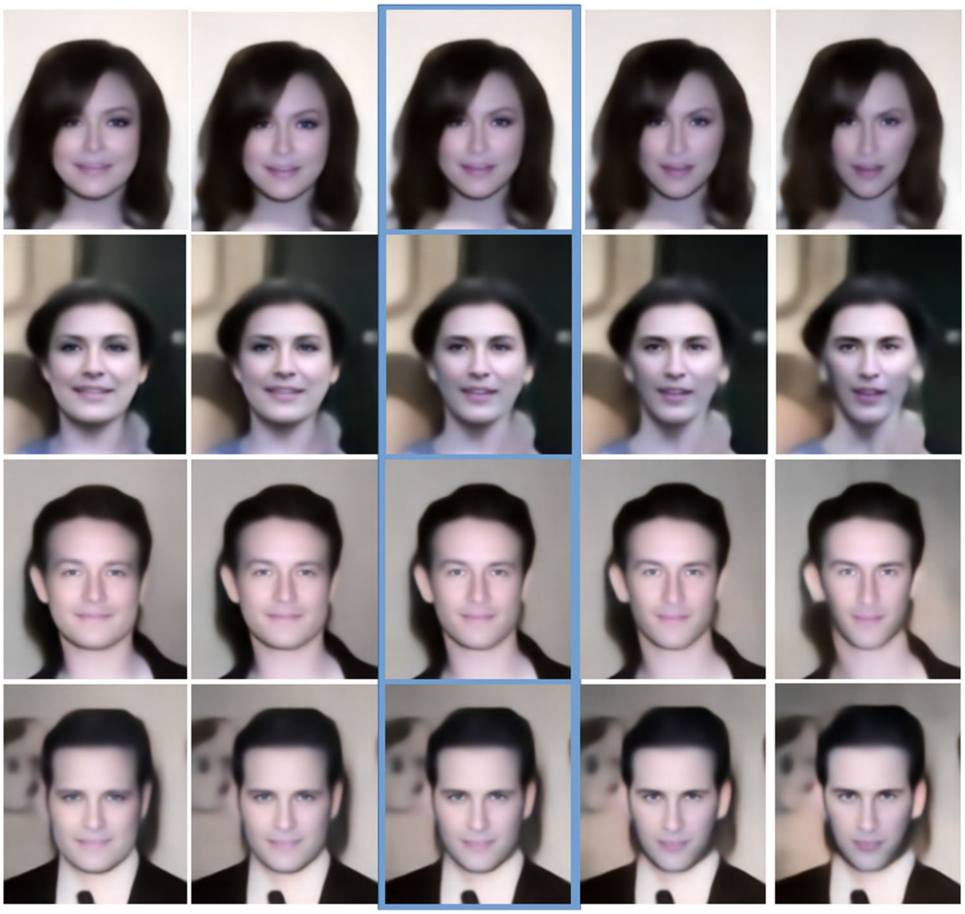

Table 7 Mean relative errors e_r in percent (with 99% confidence intervals) for four different aspect ratio modifications and four different sizes of CAE's latent space

	Alteration magnitude			
	− 20%	− 10%	10%	20%
1024	27.2 ± 0.3	13.4 ± 0.2	13.1 ± 0.2	27.5 ± 0.3
2048	26.8 ± 0.3	13.0 ± 0.2	12.9 ± 0.2	26.9 ± 0.3
4096	27.5 ± 0.3	13.5 ± 0.2	12.9 ± 0.2	26.4 ± 0.3
8192	26.5 ± 0.3	13.3 ± 0.2	12.5 ± 0.2	26.5 ± 0.3

The major perceived limitation of the architecture is its linearity. SPCA analysis, as a linear transformation, cannot capture nonlinear relationships, which may hold for many high-level object descriptors. Also, nonlinear features might be necessary for decoupling representations for different attributes. Therefore, it could be necessary to elaborate methods for implementing nonlinear mappings between latent and attribute spaces.

There are many other issues that need to be elaborated to improve robustness, versatility and performance of the proposed methodology for visual objects appearance manipulation. These include for example, research on development of quantitative descriptors that correctly reflect more complex high-level visual features. Accurate quantitative description of high-level features is a difficult, nonlinear regression problem. Elaboration of methods for accurate high-level visual attribute tagging, which is essential for performing latent-to-attribute space transformation, would enable application of the proposed algorithm to a variety of complex image editing tasks.

Acknowledgements This research was funded by the National Centre for Research and Development under the Grant CYBERSECIDENT/382354/II/NCBR/2018.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest in relation to this article.

References

1. Antipov G, Baccouche M, Dugelay J (2017) Face aging with conditional generative adversarial networks. arXiv:1702.01983
2. Baek K, Bang D, Shim H (2018) Editable generative adversarial networks: generating and editing faces simultaneously. arXiv:1807.07700
3. Barshan E, Ghodsi A, Azimifar Z, Zolghadri Jahromi M (2011) Supervised principal component analysis: visualization, classification and regression on subspaces and submanifolds. Pattern Recognit 44(7):1357–1371
4. Bengio Y, Thibodeau-Laufer E, Yosinski J (2013) Deep generative stochastic networks trainable by backprop. arXiv:1306.1091
5. Bodnar C (2018) Text to image synthesis using generative adversarial networks. arXiv:1805.00676
6. Brock A, Donahue J, Simonyan K (2018) Large scale GAN training for high fidelity natural image synthesis. arXiv:1809.11096
7. Brock A, Lim T, Ritchie JM, Weston N (2016) Neural photo editing with introspective adversarial networks. arXiv:1609.07093
8. Goodfellow IJ (2017) NIPS 2016 tutorial: Generative adversarial networks. arXiv:1701.00160
9. Gorijala M, Dukkipati A (2017) Image generation and editing with variational info generative adversarial networks. arXiv:1701.04568
10. Gregor K, Danihelka I, Graves A, Rezende D, Wierstra D (2015) Draw: a recurrent neural network for image generation. In: Proceedings of the 32nd international conference on machine learning, volume 37 of proceedings of machine learning research, pp 1462–1471
11. He H, Yu PS, Wang C (2018) An introduction to image synthesis with generative adversarial nets. arXiv:1803.04469
12. He Z, Zuo W, Kan M, Shan S, Chen X (2019) Attgan: facial attribute editing by only changing what you want. IEEE Trans Image Process 28(11):5464–5478
13. He Z, Zuo W, Kan M, Shan S, Chen X (2020) Attgan: facial attribute editing by only changing what you want—Tensorflow implementation. https://github.com/LynnHo/AttGAN-Tensorflow
14. Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 3128–3137
15. King DE (2009) Dlib-ml: a machine learning toolkit. J Mach Learn Res 10:1755–1758
16. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv:1312.6114
17. Lample G, Zeghidour N, Usunier N, Bordes A, Denoyer L, Ranzato M (2017) Fader networks: manipulating images by sliding attributes. arXiv:1706.00409
18. LeCun Y, Cortes C (2010) MNIST handwritten digit database
19. Masci J, Meier U, Cireşan D, Schmidhuber J (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. In: Artificial neural networks and machine learning—ICANN 2011, Lecture Notes in Computer Science, vol 6791, pp 52–59
20. Mehri S, Kumar K, Gulrajani I, Kumar R, Jain S, Sotelo J, Courville AC, Bengio Y (2016) SampleRNN: an unconditional end-to-end neural audio generation model. arXiv:1612.07837
21. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv:1411.1784
22. Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) Wavenet: a generative model for raw audio. arXiv:1609.03499
23. Paulus R, Xiong C, Socher R (2017) A deep reinforced model for abstractive summarization. arXiv:1705.04304
24. Perarnau G, van de Weijer J, Raducanu B, Álvarez JM (2016) Invertible conditional gans for image editing. arXiv:1611.06355
25. Pieters M, Wiering M (2018) Comparing generative adversarial network techniques for image creation and modification. arXiv:1803.09093
26. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434
27. Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis. arXiv:1605.05396
28. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville AC (2017) Char2wav: end-to-end speech synthesis
29. Sutskever I, Vinyals D, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the 27th international conference on neural information processing systems, vol 2, NIPS'14, pp 3104–3112
30. van den Oord Aä, Kalchbrenner N, Kavukcuoglu K (2015) Pixel recurrent neural networks. arXiv:1601.06759
31. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: Proceedings of the 32nd international conference on machine learning, volume 37 of proceedings of machine learning research, pp 2048–2057
32. Xu T, Zhang P, Huang Q, Zhang H, Gan Z, Huang X, He X (2017) AttnGAN: fine-grained text to image generation with attentional generative adversarial networks. arXiv:1711.10485
33. Yan X, Yang J, Sohn K, Lee H (2016) Attribute2Image: conditional image generation from visual attributes. In: European conference on computer vision
34. Yang S, Luo P, Loy CC, Tang X (2015) From facial parts responses to face detection: a deep learning approach. In: IEEE international conference on computer vision (ICCV)
35. Zhang H, Xu T, Li H, Zhang S, Wang X, Huang X, Metaxas D (2017) Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. arXiv:1612.03242