Making Sense of CNNs: Interpreting Deep Representations & Their Invariances with INNs

Robin Rombach*, Patrick Esser*, and Björn Ommer

Interdisciplinary Center for Scientific Computing, HCI, Heidelberg University
https://hci.iwr.uni-heidelberg.de/compvis

Abstract. To tackle increasingly complex tasks, it has become an essential ability of neural networks to learn abstract representations. These task-specific representations and, particularly, the invariances they capture turn neural networks into black box models that lack interpretability. To open such a black box, it is, therefore, crucial to uncover the different semantic concepts a model has learned as well as those that it has learned to be invariant to. We present an approach based on INNs that (i) recovers the task-specific, learned invariances by disentangling the remaining factor of variation in the data and that (ii) invertibly transforms these recovered invariances combined with the model representation into an equally expressive one with accessible semantic concepts. As a consequence, neural network representations become understandable by providing the means to (i) expose their semantic meaning, (ii) semantically modify a representation, and (iii) visualize individual learned semantic concepts and invariances. Our invertible approach significantly extends the abilities to understand black box models by enabling post-hoc interpretations of state-of-the-art networks without compromising their performance.

1 Introduction

Key to the wide success of deep neural networks is end-to-end learning of powerful hidden representations that aim to *(i)* capture all task-relevant characteristics while *(ii)* being invariant to all other variability in the data [32, 1]. Deep learning can yield abstract representations that are perfectly adapted feature encodings for the task at hand. However, their increasing abstraction capability and performance comes at the expense of a lack in interpretability [3]: Although the network may solve a problem, it does not convey an understanding of its predictions or their causes, oftentimes leaving the impression of a black box [40]. In particular, users are missing an explanation of semantic concepts that the model has learned to *represent* and of those it has learned to *ignore*, i.e. its invariances.

Providing such explanations and an understanding of network predictions and their causes is thus crucial for transparent AI. Not only is this relevant to discover limitations and promising directions for future improvements of the

* Both authors contributed equally to this work.

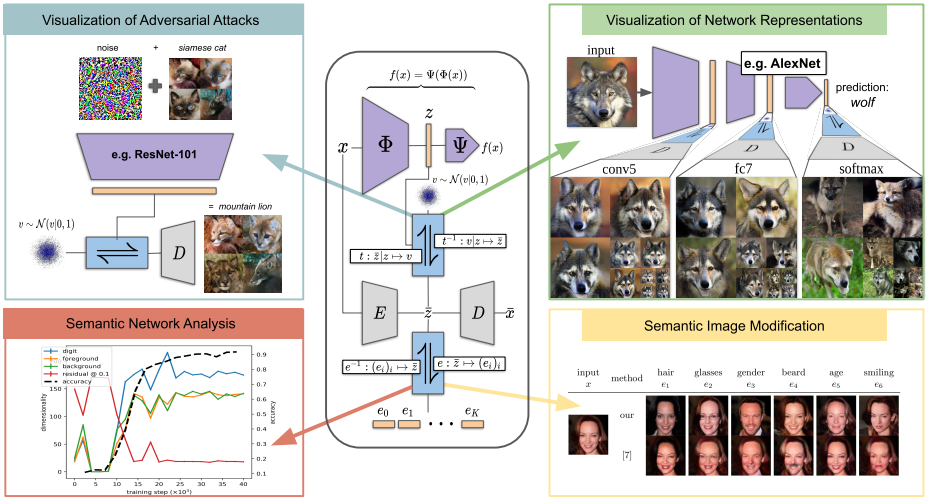


Fig. 1. Proposed architecture. We provide post-hoc interpretation for a given deep network $f = \Psi \circ \Phi$. For a deep representation $z = \Phi(x)$ a conditional INN t recovers Φ's invariances v from a representation $\bar{z}$ which contains entangled information about *both* z and v. The INN e then translates the representation $\bar{z}$ into a factorized representation with accessible semantic concepts. This approach allows for various applications, including visualizations of network representations of natural and altered inputs, semantic network analysis and semantic image modifications.

AI system itself, but also for compliance with legislation [21, 9], knowledge distillation from such a system [34], and post-hoc verification of the model [50]. Consequently, research on interpretable deep models has recently gained a lot of attention, particularly methods that investigate latent representations to understand what the model has learned [50, 58, 4, 16, 15].

Challenges & aims Assessing these latent representations is challenging due to two fundamental issues: *(i)* to achieve robustness and generalization despite noisy inputs and data variability, hidden layers exhibit a distributed coding of semantically meaningful concepts [17]. Attributing semantics to a single neuron via backpropagation [41] or synthesis [62] is thus impossible without altering the network [42, 67], which typically degrades performance. *(ii)* end-to-end learning trains deep representations towards a goal task, making them invariant to features irrelevant for this goal. Understanding these characteristics that a representation has abstracted away is challenging, since we essentially need to portray features that have been discarded.

These challenges call for a method that can interpret existing network representations by recovering their invariances without modifying them. Given these recovered invariances, we seek an invertible mapping that translates a representation *and* the invariances onto understandable semantic concepts. The mapping disentangles the distributed encoding of the high-dimensional representation and its invariances by projecting them onto separate multi-dimensional factors that

correspond to human understandable semantic concepts. Both this translation and the recovering of invariances are implemented with invertible neural networks (INNs) [47, 12, 27]. For the translation, this guarantees that the resulting understandable representation is equally expressive as the model representation combined with the recovered invariances (no information is lost). Its invertibility also warrants that feature modifications applied in the semantic domain correctly adjust the recovered representation.

Our contributions to a comprehensive understanding of deep representations are as follows: (i) We present an approach, which, by utilizing invertible neural networks, improves the understanding of representations produced by existing network architectures with no need for re-training or otherwise compromising their performance. (ii) Our generative approach is able to recover the invariances that result from the non-injective projection (of input onto a latent representation) which deep networks typically learn. This model then provides a probabilistic visualization of the latent representation and its invariances. (iii) We bijectively translate an arbitrarily abstract representation and its invariances via a non-linear transformation into another representation of equal expressiveness, but with accessible semantic concepts. (iv) The invertibility also enables manipulation of the original latent representations in a semantically understandable manner, thus facilitating further diagnostics of a network.

2 Background

Two main approaches to interpretable AI can be identified, those which aim to incorporate interpretability directly into the design of models, and those which aim to provide interpretability to existing models [42]. Approaches from the first category range from modifications of network architectures [67], over regularization of models encouraging interpretability [38, 46], towards combinations of both [64]. However, these approaches always involve a trade-off between model performance and model interpretability. Being of the latter category, our approach allows to interpret representations of existing models without compromising their performance.

To better understand what an existing model has learned, its representations must be studied [50]. [58] shows that both random directions and coordinate axes in the feature space of networks can represent semantic properties and concludes that they are not necessarily represented by individual neurons. Different works attempt to select groups of neurons which have a certain semantic meaning, such as based on scenes [66], objects [55] and object parts [56]. [4] studied the interpretability of neurons, and found that a rotation of the representation space spanned by the neurons decreases its interpretability. While this suggests that the neurons provide a more interpretable basis compared to a random basis, [16] shows that the choice of basis is not the only challenge for interpretability of representations. Their findings demonstrate that learned representations are distributed, *i.e.* a single semantic concept is encoded by an activation pattern involving multiple neurons, and a single neuron is involved in the encoding of

multiple different semantic concepts. Instead of selecting a set of neurons directly, [15] learns an INN that transforms the original representation space to an interpretable space, where a single semantic concept is represented by a known group of neurons and a single neuron is involved in the encoding of just a single semantic concept. However, to interpret not only the representation itself but also its invariances, it is insufficient to transform only the representation itself. Our approach therefore transforms the latent representation space of an autoencoder, which has the capacity to represent its inputs faithfully, and subsequently translates a model representation and its invariances into this space for semantic interpretation and visualization.

A large body of works approach interpretability of existing networks based on visualizations. [53] uses gradients of network outputs with respect to a convolutional layer to obtain coarse localization maps. [3] proposes an approach to obtain pixel-wise relevance scores for a specific class of models which is generalized in [41]. To obtain richer visual interpretations, [63, 57, 62, 39] reconstruct images which maximally activate certain neurons. [45] uses a generator network for this task, which was introduced in [13] for reconstructing images from their feature representation. Our key insight is that these existing approaches do not explicitly account for the invariances learned by a model. Invariances imply that feature inversion is a one-to-many mapping and thus they must be recovered to solve the task. Recently, [54] introduced a GAN-based approach that utilizes features of a pre-trained classifier as a semantic pyramid for image generation. [44] used samples from an autoregressive model of images conditioned on a feature representation to gain insights into the representation's invariances. In contrast, our approach recovers an explicit representation of the invariances, which can be recombined with modified feature representations, and thus makes the effect of modifications to representations, *e.g.* through adversarial attacks, visible.

Other works consider visual interpretations for specialized models. [51] showed that the quality of images which maximally activate certain neurons is significantly improved when activating neurons of an adversarially robust classifer. [5] explores the relationship between neurons and the images produced by a Generative Adversarial Network. For the same class of models, [19] finds directions in their input space which represent semantic concepts corresponding to certain cognitive properties. Such semantic directions have previously also been found in classifier networks [59] but requires aligned data. All of these approaches require either special training of models, are limited to a very special class of models which already provide visualizations or depend on special assumptions on model and data. In contrast, our approach can be applied to arbitrary models without re-training or modifying them, and provides both visualizations and semantic explanations, for both the model's representation and its learned invariances.

3 Approach

Common tasks of computer vision can be phrased as a mapping from an input image x to some output $f(x)$ such as a classification of the image, a regression

(e.g. of object locations), a (semantic) segmentation map, or a re-synthesis that yields another image. Deep learning utilizes a hierarchy of intermediate network layers that gradually transform the input into increasingly more abstract representations. Let $z = \Phi(x) \in \mathbb{R}^{N_z}$ be the representation extracted by one such layer (without loss of generality we consider z to be a N_z-dim vector, flattening it if necessary) and $f(x) = \Psi(z) = \Psi(\Phi(x))$ the mapping onto the output.

An essential characteristic of a deep feature encoding z is the increasing abstractness of higher feature encoding layers and the resulting reduction of information. To explain a latent representation, we need to recover its invariances v and make z *and* v interpretable by learning a bijective mapping onto understandable semantic concepts, see Fig. 1. Sec. 3.1 describes our INN t to recover an encoding v of the invariances. Due to the generative nature of t, our approach can correctly sample visualizations of the model representation and its invariances without leaving the underlying data distribution and introducing artifacts. With v then available, Sec. 3.2 presents an INN e that translates t's encoding of z and v without losing information onto disentangled semantic concepts. Moreover, the invertibility allows modifications in the semantic domain to correctly project back onto the original representation or into image space.

3.1 Recovering the Invariances of Deep Models

Learning an Encoding to Help Recover Invariances Key to a deep representation is not only the information z captures, but also what is has learned to abstract away. To learn what z misses with respect to x, we need an encoding $\bar{z}$, which, in contrast to z, includes these invariances. Without making prior assumptions about the deep model f, autoencoders provide a generic way to obtain such an encoding $\bar{z}$, since they ensure that their input x can be recovered from their learned representation $\bar{z}$, which hence also comprises the invariances.

Therefore, we learn an autoencoder with an encoder E that provides the data representation $\bar{z} = E(x)$ and a decoder D producing the data reconstruction $\bar{x} = D(\bar{z})$. Sec. 3.2 will utilize the decoding from $\bar{z}$ to $\bar{x}$ to visualize both z and v. The autoencoder is trained to reconstruct its inputs by minimizing a perceptual metric between input and reconstruction, $\|x - \bar{x}\|$, as in [13]. The details of the architecture and training procedure can be found in Sec. A.1. It is crucial that the autoencoder only needs to be trained once on the training data. Consequently, the same E can be used to interpret different representations z, *e.g.* different models or layers within a model, thus ensuring fair comparisons between them. Moreover, the complexity of the autoencoder can be adjusted based on the computational needs, allowing us to work with much lower dimensional encodings $\bar{z}$ compared to reconstructing the invariances directly from the images x. This reduces the computational demands of our approach significantly.

Learning a Conditional INN that Recovers Invariances Due to the reconstruction task of the autoencoder, $\bar{z}$ not only contains the invariances v, but also the representation z. Thus, we must disentangle [14, 33, 28] v and z using a mapping $t(\cdot|z) : \bar{z} \mapsto v = t(\bar{z}|z)$ which, depending on z, extracts v from $\bar{z}$.

Besides extracting the invariances from a given $\bar{z}$, t must also enable an inverse mapping from given model representations z to $\bar{z}$ to support a further mapping onto semantic concepts (Sec. 3.2) and visualization based on $D(\bar{z})$. There are many different x with $\Phi(x) = z$, namely all those x which differ only in properties that Φ is invariant to. Thus, there are also many different $\bar{z}$ that this mapping must recover. Consequently, the mapping from z to $\bar{z}$ is set-valued. However, to understand f we do not want to recover all possible $\bar{z}$, but only those which are likely under the training distribution of the autoencoder. In particular, this excludes unnatural images such as those obtained by DeepDream [43], or adversarial attacks [58]. In conclusion, we need to sample $\bar{z} \sim p(\bar{z}|z)$.

To avoid a costly inversion process of Φ, t must be invertible (implemented as an INN) so that a change of variables

$$p(\bar{z}|z) = \frac{p(v|z)}{|\det \nabla(t^{-1})(v|z)|} \quad \text{where } v = t(\bar{z}|z) \tag{1}$$

yields $p(\bar{z}|z)$ by means of the distribution $p(v|z)$ of invariances, given a model representation z. Here, the denominator denotes the absolute value of the determinant of Jacobian $\nabla(t^{-1})$ of $v \mapsto t^{-1}(v|z) = \bar{z}$, which is efficient to compute for common invertible network architectures. Consequently, we obtain $\bar{z}$ for given z by sampling from the invariant space v given z and then applying t^{-1},

$$\bar{z} \sim p(\bar{z}|z) \quad \Longleftrightarrow \quad v \sim p(v|z), \bar{z} = t^{-1}(v|z). \tag{2}$$

Since v is the invariant space for z, both are complementary thus implying independence $p(v|z) = p(v)$. Because a powerful transformation t^{-1} can transform between two arbitrary densities, we can assume without loss of generality a Gaussian prior $p(v) = \mathcal{N}(v|0, \mathbb{1})$. Given this prior, our task is then to learn the transformation t that maps $\mathcal{N}(v|0, \mathbb{1})$ onto $p(\bar{z}|z)$. To this end, we maximize the log-likelihood of $\bar{z}$ given z, which results in a per-example loss of

$$\ell(\bar{z}, z) = -\log p(\bar{z}|z) = -\log \mathcal{N}(t(\bar{z}|z)) - \log|\det \nabla t(\bar{z}|z)|. \tag{3}$$

Minimizing this loss over the training data distribution $p(x)$ gives t, a bijective mapping between $\bar{z}$ and (z, v),

$$\mathcal{L}(t) = \mathbb{E}_{x\sim p(x)} \left[\ell(E(x), \Phi(x))\right] \tag{4}$$

$$= \mathbb{E}_{x\sim p(x)} \left[\frac{1}{2}\|t(E(x)|\Phi(x))\|^2 + N_{\bar{z}} \log 2\pi - \log|\det \nabla t(E(x)|\Phi(x))|\right] \tag{5}$$

Note that both E and Φ remain fixed during minimization of $\mathcal{L}$.

3.2 Interpreting Representations and Their Invariances

Visualizing Representations and Invariances For an image representation $z = \Phi(x)$, Eq. (2) presents an efficient approach (a single forward pass through the INN t) to sample an encoding $\bar{z}$, which is a combination of z with a particular realization of its invariances v. Sampling multiple realizations of $\bar{z}$ for

a given z highlights what remains constant and what changes due to different v: information preserved in the representation z remains constant over different samples and information discarded by the model ends up in the invariances v and shows changes over different samples. Visualizing the samples $\bar{z} \sim p(\bar{z}|z)$ with $\bar{x} = D(\bar{z})$ portrays this constancy and changes due to different v. To complement this visualization, in the following, we learn a transformation of $\bar{z}$ into a semantically meaningful representation which allows to uncover the semantics captured by z and v.

Learning an INN to Produce Semantic Interpretations The autoencoder representation $\bar{z}$ is an equivalent representation of (z, v) but its feature dimensions do not necessarily correspond to semantic concepts [17]. More generally, without supervision, we cannot reliably discover semantically meaningful, explanatory factors of $\bar{z}$ [37]. In order to explain $\bar{z}$ in terms of given semantic concepts, we apply the approach of [15] and learn a bijective transformation of $\bar{z}$ to an interpretable representation $e(\bar{z})$ where different groups of components, called factors, correspond to semantic concepts.

To learn the transformation e, we parameterize e by an INN and assume that semantic concepts are defined implicitly by pairs of images, *i.e.* for each semantic concept we have access to training pairs x^a, x^b that have the respective concept in common. For example, the semantic concept 'smiling' is defined by pairs of images, where either both images show smiling persons or both images show non-smiling persons. Applying this formulation, input pairs which are similar in a certain semantic concept are similar in the corresponding factor of the interpretable representation $e(\bar{z})$.

Following [15], the loss for training the invertible network e is then given by

$$\begin{aligned}\mathcal{L}(e) = \mathbb{E}_{x^a, x^b}\big[&-\log p(e(E(x^a)), e(E(x^b))) \\ &- \log|\det \nabla e(E(x^a))| - \log|\det \nabla e(E(x^b))|\big]\,. \end{aligned} \tag{6}$$

Further details regarding the application of this approach within our setting can be found in the supplementary, Sec. A.2.

Interpretation by Applying the Learned INNs After training, the combination of e with t from Sec. 3.1 provides semantic interpretations given a model representation z: Eq. (2) gives realizations of the invariances v which are combined with z to produce $\bar{z} = t^{-1}(v|z)$. Then e transforms $\bar{z}$ without loss of information into a semantically accessible representation $(e_i)_i = e(\bar{z}) = e(t^{-1}(v|z))$ consisting of different semantic factors e_i. Comparing the e_i for different model representations z and invariances v allows us to observe which semantic concepts the model representation $z = \Phi(\cdot)$ is sensitive to, and which it is invariant to.

Semantic Modifications of Latent Representations t^{-1} and e not only interpret a representation z in terms of accessible semantic concepts $(e_i)_i$. Given

Table 1. FID scores for layer visualizations of *AlexNet*, obtained with our method and [13] (D&B). Scores are calculated on the *Animals* dataset.

layer	conv5	fc6	fc7	fc8	output
ours	$\mathbf{23.6 \pm 0.5}$	$\mathbf{24.3 \pm 0.7}$	$\mathbf{24.9 \pm 0.4}$	$\mathbf{26.4 \pm 0.4}$	$\mathbf{27.4 \pm 0.3}$
D&B	25.2	**24.9**	27.2	36.1	352.6

$v \sim p(v)$, they also allow to modify $\bar{z} = t^{-1}(v|z)$ in a semantically meaningful manner by altering its corresponding $(e_i)_i$ and then applying the inverse translation e^{-1},

$$\bar{z} \xrightarrow{e} (e_i) \xrightarrow{\text{modification}} (e_i^*) \xrightarrow{e^{-1}} \bar{z}^* \tag{7}$$

The modified representation $\bar{z}^*$ is then readily transformed back into image space $\bar{x}^* = D(\bar{z}^*)$. Besides visual interpretation of the modification, $\bar{x}^*$ can be fed into the model $\Psi(\Phi(\bar{x}^*))$ to probe for sensitivity to certain semantic concepts.

4 Experiments

To explore the applicability of our approach, we conduct experiments on several models which we aim to understand: *SqueezeNet* [24], which provides lightweight classification, *FaceNet* [52], a baseline for face recognition and clustering, trained on the *VGGFace2 dataset* [7], and variants *ResNet* [22], a popular architecture, often used when finetuning a classifier on a specific task and dataset.

Experiments are conducted on the following datasets: *CelebA* [36], *AnimalFaces* [35], *Animals* (containing carnivorous animals, see Sec. B.3), *ImageNet* [11] and *ColorMNIST*, which is an augmented version of the *MNIST* dataset [31], where both background and foreground have random, independent colors.

4.1 Comparison to Existing Methods

A key insight of our work is that reconstructions from a given model's representation $z = \Phi(x)$ are impossible if the invariances the model has learned are not considered. In Fig. 2 we compare to existing methods that either try to reconstruct the image via gradient-based optimization [39] or by training a reconstruction network directly on the representations z [13]. By conditionally sampling images $\bar{x} = D(\bar{z})$, where we obtain $\bar{z}$ via the INN t as described in Eq. (2) based on the invariances $v \sim p(v) = \mathcal{N}(0, \mathbb{1})$, we bypass this shortcoming and obtain natural images without artifacts for any layer depth. The increased image quality is further confirmed by the FID scores reported in Tab. 1.

4.2 Understanding Models

Interpreting a Face Recognition Model *FaceNet* [52] is a widely accepted baseline in the field of face recognition. This model embeds input images of human faces into a latent space where similar images have a small L_2-distance. We

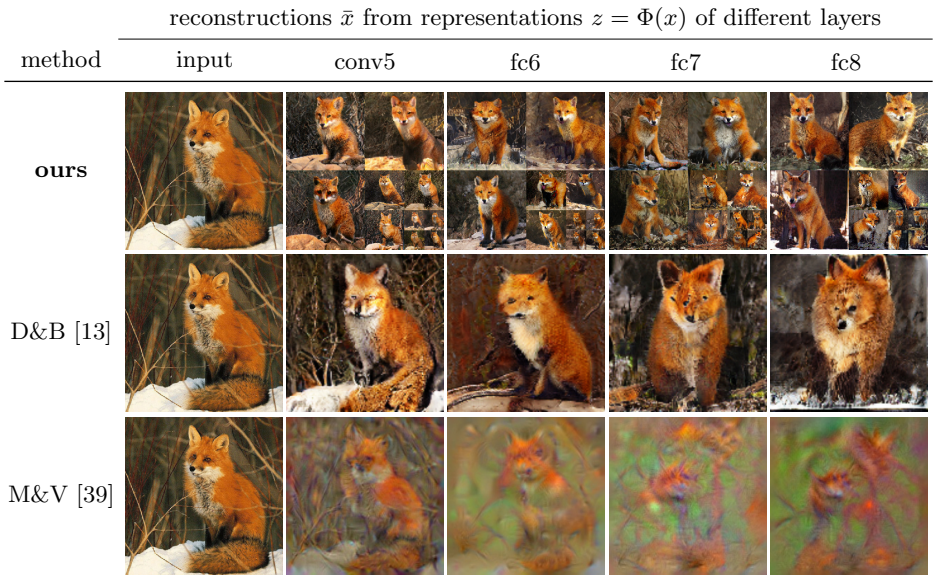


Fig. 2. Comparison to existing network inversion methods for *AlexNet* [29]. In contrast to the methods of [13] (D&B) and [39] (M&V), our invertible method explicitly samples the invariances of Φ w.r.t. the data, which circumvents a common cause for artifacts and produces natural images independent of the depth of the layer which is reconstructed.

aim to understand the process of face recognition within this model by analyzing and visualizing learned invariances for several layers explicitly; see Tab. S12 for a detailed breakdown of the various layers of *FaceNet*. For the experiment, we use a pretrained *FaceNet* and train the generative model presented in Eq. (2) by conditioning on various layers. Fig. 3 depicts the amount of variance present in each selected layer when generating $n = 250$ samples for each of 100 different input images. This variance serves as a proxy for the amount of abstraction capability *FaceNet* has learned in its respective layers: More abstract representations allow for a rich variety of corresponding synthesized images, which results in a large variance in image space when being decoded. We observe an approximate exponential growth of learned invariances with increasing layer depth, suggesting that abstraction mainly happens in the deepest layers of the network. Furthermore, we are able to synthesize images that correspond to the given model representation for each selected layer.

How Does Relevance of Different Concepts Emerge During Training? Humans tend to provide explanations of entities by describing them in terms of their semantics, e.g. size or color. In a similar fashion, we want to semantically understand how a network (here: *SqueezeNet*) learns to solve a given problem. Intuitively, a network should for example be able to solve a given classification problem by focusing on the relevant information while discarding task-irrelevant

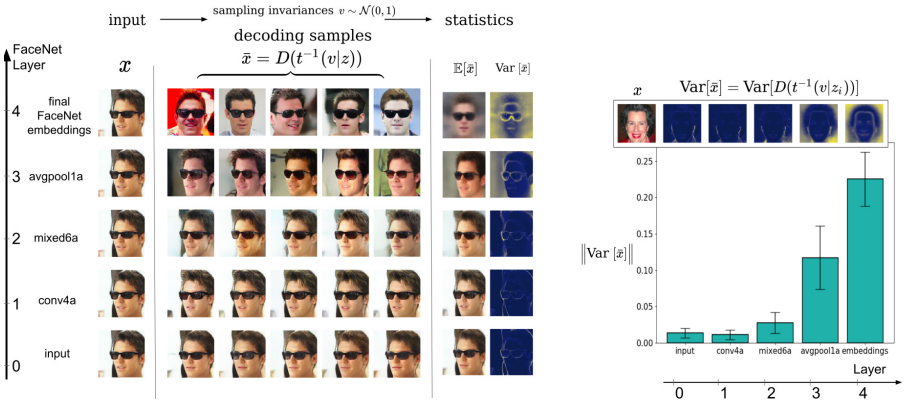


Fig. 3. *left*: Visualizing *FaceNet* representations and their invariances. Sampling multiple reconstructions $\bar{x} = D(t^{-1}(v|z))$ shows the degree of invariance learned by different layers. The invariance w.r.t. pose increases for deeper layers as expected for face identification. Surprisingly, FaceNet uses glasses as an identity feature throughout all its layers as evident from the spatial mean and variance plots, where the glasses are still visible. This reveals a bias and weakness of the model. *right*: Spatially averaged variances over multiple x for different layers.

information. To build on this intuition, we construct a toy problem: Digit classification on ColorMNIST. We expect the model to ignore both the random back- and foreground color of the input data, as it does not help making a classification decision. Thus, we apply the invertible approach presented in Sec. 3.2 and recover three distinct factors: *digit class*, *background color* and *foreground color*. To capture the semantic changes occuring over the course of training of this classifier, we couple 20 instances of the invertible interpretation model on the last convolutional layer, each representing a checkpoint between iteration 0 and iteration 40000 (equally distributed). The result is shown in Fig. 4: We see that the *digit* factor becomes increasingly more relevant, with its relevance being strongly correlated to the accuracy of the model.

4.3 Effects of Data Shifts on Models

This section investigates the effects that altering the input data has on the model we want to understand. We examine these effects by manipulating the input data explicitly through adversarial attacks or image stylization.

How Do Adversarial Attacks Affect Network Representations? Here, we experiment with *Fast Gradient Sign* (FGSM) attacks [20], which manipulate the input image by maximizing the objective of a given classification model. To understand how such an attack modifies representations of a given model, we first compute the image's invariances with respect to the model as $v = t(E(x)|\Phi(x))$. For an attacked image x^*, we then compute the attacked representation as $z^* =$

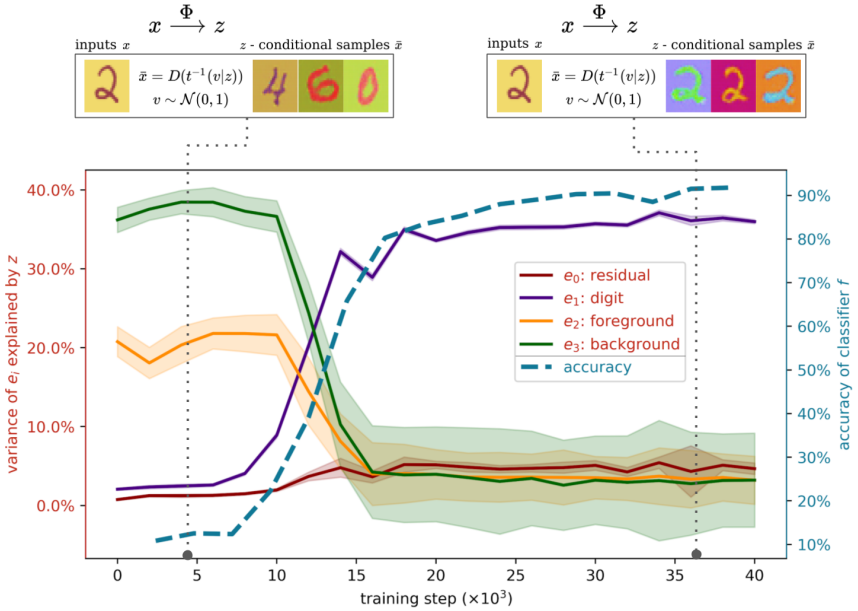


Fig. 4. Analyzing the degree to which different semantic concepts are captured by a network representation changes as training progresses. For *SqueezeNet* on *ColorMNIST* we measure how much the data varies in different semantic concepts e_i and how much of this variability is captured by z at different training iterations. Early on z is sensitive to foreground and background color, and later on it learns to focus on the digit attribute. The ability to encode this semantic concept is proportional to the classification accuracy achieved by z. At training iterations 4k and 36k we apply our method to visualize model representations and thereby illustrate how their content changes during training.

$\Phi(x^*)$. Decoding this representation with the original invariance v, allows us to precisely visualize what the adversarial attack changed. This decoding, $\bar{x}^* = D(t(v|z^*))$, is shown in Fig. 5. We observe that, over layers of the network, the adversarial attack gradually changes the representation towards its target. Its ability to do so is strongly correlated with the amount of invariances, quantified as the total variance explained by v (see Sec. B.2), for a given layer as also observed in [25]. For additional examples, see Fig. S13.

How Does Training on Different Data Affect the Model? [18] proposed the hypothesis that classification networks based on convolutional blocks mainly focus on texture patterns to obtain class probabilities. We further validate this hypothesis by training our invertible network t conditioned on pre-logits $z = \Phi(x)$ (*i.e.* the penultimate layer) of two ResNet-50 realizations. As shown in Fig. 6, a ResNet architecture trained on standard ImageNet is susceptible to the so-called "texture-bias", as samples generated conditioned on representation of pure texture images consistently show valid images of corresponding input

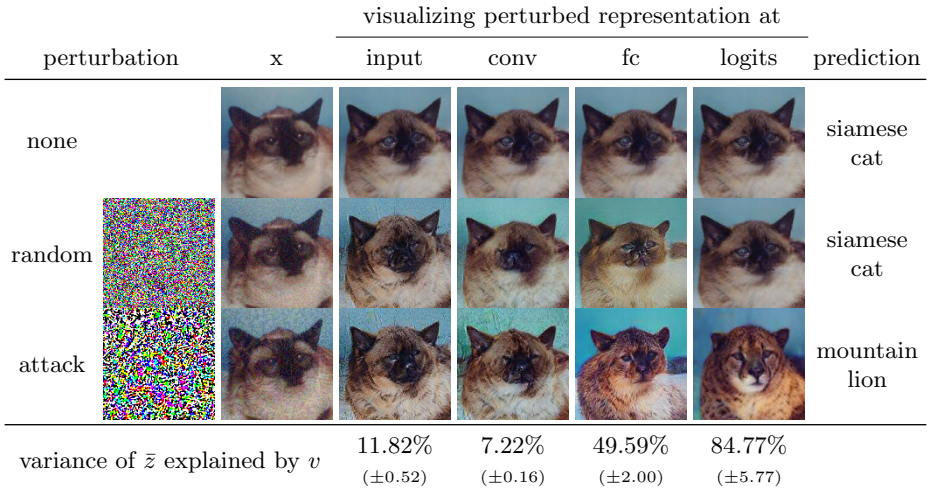


Fig. 5. Visualizing FGSM adversarial attacks on *ResNet-101*. To the human eye, the original image and its attacked version are almost indistinguishable. However, the input image is correctly classified as "siamese cat", while the attacked version is classified as "mountain lion". Our approach visualizes how the attack spreads throughout the network. Reconstructions of representations of attacked images demonstrate that the attack targets the semantic content of deep layers. The variance of $\bar{z}$ explained by v combined with these visualizations show how increasing invariances cause vulnerability to adversarial attacks.

classes. We furthermore visualize that this behavior can indeed be removed by training the same architecture on a stylized version of ImageNet [1]; the classifier does focus on shape. Rows 10-12 of Fig. 6 show that the proposed approach can be used to generate sketch-based content with the texture-agnostic network.

4.4 Modifying Representations

Invertible access to semantic concepts enables targeted modifications of representations $\bar{z}$. In combination with a decoder for $\bar{z}$, we obtain semantic image editing capabilities. We provide an example in Fig. 7, where we modify the factors hair color, glasses, gender, beard, age and smile. We infer $\bar{z} = E(x)$ from an input image. Our semantic INN e then translates this representation into semantic factors $(e_i)_i = e(\bar{z})$, where individual semantic concepts can be modified independently via the corresponding factor e_i. In particular, we can replace each factor with that from another image, effectively transferring semantics from one representation onto another. Due to the invertibility of e, the modified representation can be translated back into the space of the autoencoder and is readily decoded to a modified image x^*. Additional examples can be found in Sec. B.5.

[1] we used weights available at `https://github.com/rgeirhos/texture-vs-shape`

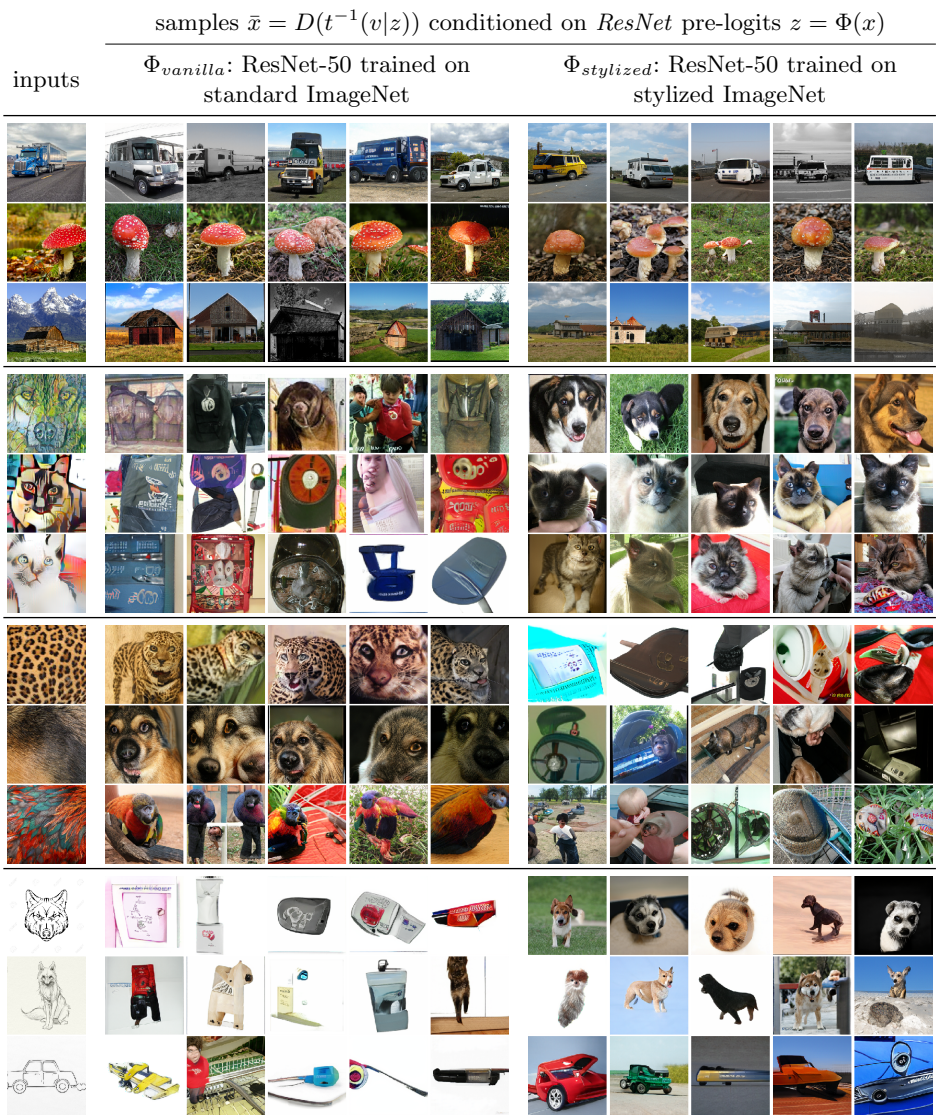
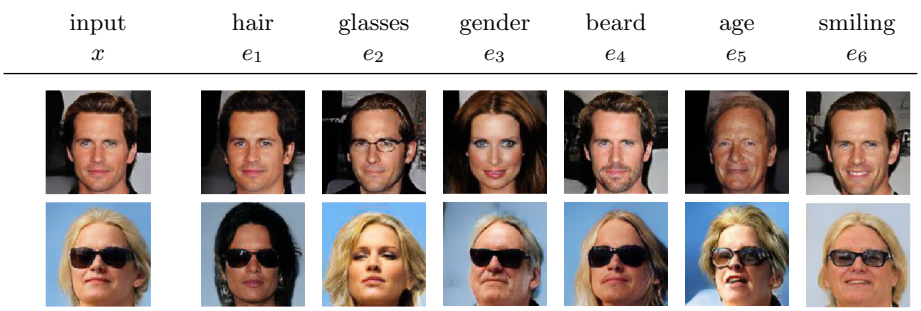


Fig. 6. Revealing texture bias in ImageNet classifiers. We compare visualizations of z from the penultimate layer of *ResNet-50* trained on standard ImageNet (left) and a stylized version of ImageNet (right). On natural images (rows 1-3) both models recognize the input, removing textures through stylization (rows 4-6) makes images unrecognizable to the standard model, however it recognizes objects from textured patches (rows 7-9). Rows 10-12 show that a model without texture bias can be used for sketch-to-image synthesis.



mean embedding distance (± std)	0.872 (±0.048)	1.000 (±0.046)	1.061 (±0.030)	0.803 (±0.041)	0.874 (±0.053)	0.833 (±0.034)

Fig. 7. Semantic Modifications on CelebA. For each column, after inferring the semantic factors $(e_i)_i = e(E(x))$ of the input x, we replace one factor e_i by that from another randomly chosen image that differs in this concept. The inverse of e translates this semantic change back into a modified $\bar{z}$, which is decoded to a semantically modified image. Distances between *FaceNet* embeddings before and after modification demonstrate its sensitivity to differences in gender and glasses (see also Fig. 3).

To observe which semantic concepts *FaceNet* is sensitive to, we compute the average distance $\|f(x) - f(x^*)\|$ between its embeddings of x and semantically modified x^* over the test set (last row in Fig. 7). Evidently, FaceNet is particularly sensitive to differences in gender and glasses. The latter suggests a failure of FaceNet to identify persons correctly after they put on glasses.

5 Conclusion

Understanding a representation in terms of both its semantics and learned invariances is crucial for interpretation of deep networks. We presented an approach to *(i)* recover the invariances a model has learned and *(ii)* translate the representation and its invariances onto an equally expressive yet semantically accessible encoding. Our diagnostic method is applicable in a plug-and-play fashion on top of existing deep models with no need to alter or retrain them. Since our translation onto semantic factors is bijective, it loses no information and also allows for semantic modifications. Moreover, recovering invariances probabilistically guarantees that we can correctly visualize representations and sample them without leaving the underlying distribution, which is a common cause for artifacts. Altogether, our approach constitutes a powerful, widely applicable diagnostic pipeline for explaining deep representations.

Acknowledgments

This work has been supported in part by the German Research Foundation (DFG) projects 371923335, 421703927, and EXC 2181/1 - 390900948 and the German federal ministry BMWi within the project “KI Absicherung”.

References

1. Achille, A., Soatto, S.: Emergence of invariance and disentanglement in deep representations. The Journal of Machine Learning Research **19**(1), 1947–1980 (2018)
2. Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E.W., Klessen, R.S., Maier-Hein, L., Rother, C., Köthe, U.: Analyzing inverse problems with invertible neural networks (2018)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
4. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jul 2017). https://doi.org/10.1109/cvpr.2017.354, http://dx.doi.org/10.1109/CVPR.2017.354
5. Bau, D., Zhu, J.Y., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A.: Gan dissection: Visualizing and understanding generative adversarial networks (2018)
6. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
7. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). pp. 67–74. IEEE (2018)
8. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
9. Commission, E.: On artificial intelligence - a european approach to excellence and trust. Tech. rep. (2020 (accessed February, 2020)), https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2020:65:FIN
10. Dai, B., Wipf, D.: Diagnosing and enhancing vae models (2019)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp (2016)
13. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks (2016)
14. Esser, P., Haux, J., Ommer, B.: Unsupervised robust disentangling of latent characteristics for image synthesis. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2019). https://doi.org/10.1109/iccv.2019.00279, http://dx.doi.org/10.1109/ICCV.2019.00279
15. Esser, P., Rombach, R., Ommer, B.: A disentangling invertible interpretation network for explaining latent representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9223–9232 (2020)
16. Fong, R., Vedaldi, A.: Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (Jun 2018). https://doi.org/10.1109/cvpr.2018.00910, http://dx.doi.org/10.1109/CVPR.2018.00910

17. Fong, R., Vedaldi, A.: Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8730–8738 (2018)
18. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)
19. Goetschalckx, L., Andonian, A., Oliva, A., Isola, P.: Ganalyze: Toward visual definitions of cognitive image properties. arXiv preprint arXiv:1906.10112 (2019)
20. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
21. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI Magazine **38**(3), 50–57 (Oct 2017). https://doi.org/10.1609/aimag.v38i3.2741, `http://dx.doi.org/10.1609/aimag.v38i3.2741`
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
23. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium (2017)
24. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arXiv preprint arXiv:1602.07360 (2016)
25. Jacobsen, J.H., Behrmann, J., Zemel, R., Bethge, M.: Excessive invariance causes adversarial vulnerability (2018)
26. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
27. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems. pp. 10215–10224 (2018)
28. Kotovenko, D., Sanakoyeu, A., Lang, S., Ommer, B.: Content and style disentanglement for artistic style transfer. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 4421–4430 (2019)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
30. Kulkarni, T.D., Whitney, W., Kohli, P., Tenenbaum, J.B.: Deep convolutional inverse graphics network (2015)
31. LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ (1998)
32. LeCun, Y.: Learning invariant feature hierarchies. In: European conference on computer vision. pp. 496–505. Springer (2012)
33. Li, Y., Singh, K.K., Ojha, U., Lee, Y.J.: Mixnmatch: Multifactor disentanglement and encoding for conditional image generation (2019)
34. Lipton, Z.C.: The mythos of model interpretability (2016)
35. Liu, M.Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., Kautz, J.: Few-shot unsupervised image-to-image translation. arXiv preprint arXiv:1905.01723 (2019)
36. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proceedings of International Conference on Computer Vision (ICCV) (December 2015)

37. Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations (2018)
38. Lorenz, D., Bereska, L., Milbich, T., Ommer, B.: Unsupervised part-based disentangling of object shape and appearance. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10947–10956 (2019)
39. Mahendran, A., Vedaldi, A.: Visualizing deep convolutional neural networks using natural pre-images. International Journal of Computer Vision **120**(3), 233–255 (2016)
40. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
41. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition **65**, 211–222 (2017)
42. Montavon, G., Samek, W., Müller, K.R.: Methods for interpreting and understanding deep neural networks. Digital Signal Processing **73**, 1–15 (2018)
43. Mordvintsev, A., Olah, C., Tyka, M.: Inceptionism: Going deeper into neural networks (2015)
44. Nash, C., Kushman, N., Williams, C.K.: Inverting supervised representations with autoregressive neural density models. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1620–1629 (2019)
45. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks (2016)
46. Plumb, G., Al-Shedivat, M., Xing, E., Talwalkar, A.: Regularizing black-box models for improved interpretability (2019)
47. Redlich, A.N.: Supervised factorial learning. Neural Computation **5**(5), 750–766 (1993). https://doi.org/10.1162/neco.1993.5.5.750
48. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on International Conference on Machine Learning-Volume 32. pp. II–1278. JMLR. org (2014)
49. Rombach, R., Esser, P., Ommer, B.: Network fusion for content creation with conditional inns (2020)
50. Samek, W., Wiegand, T., Müller, K.R.: Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296 (2017)
51. Santurkar, S., Tsipras, D., Tran, B., Ilyas, A., Engstrom, L., Madry, A.: Image synthesis with a single (robust) classifier (2019)
52. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
53. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. International Journal of Computer Vision **128**(2), 336–359 (Oct 2019). https://doi.org/10.1007/s11263-019-01228-7, http://dx.doi.org/10.1007/s11263-019-01228-7
54. Shocher, A., Gandelsman, Y., Mosseri, I., Yarom, M., Irani, M., Freeman, W.T., Dekel, T.: Semantic pyramid for image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)

55. Simon, M., Rodner, E.: Neural activation constellations: Unsupervised part model discovery with convolutional networks. 2015 IEEE International Conference on Computer Vision (ICCV) (Dec 2015). https://doi.org/10.1109/iccv.2015.136, http://dx.doi.org/10.1109/ICCV.2015.136
56. Simon, M., Rodner, E., Denzler, J.: Part detector discovery in deep convolutional neural networks. ArXiv **abs/1411.3159** (2014)
57. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
58. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2013)
59. Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., Weinberger, K.: Deep feature interpolation for image content changes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7064–7073 (2017)
60. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2251–2265 (2018)
61. Xiao, Z., Yan, Q., Amit, Y.: Generative latent flow (2019)
62. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization (2015)
63. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. Lecture Notes in Computer Science p. 818–833 (2014)
64. Zhang, Q., Nian Wu, Y., Zhu, S.C.: Interpretable convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8827–8836 (2018)
65. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018)
66. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns (2014)
67. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Jun 2016). https://doi.org/10.1109/cvpr.2016.319, http://dx.doi.org/10.1109/CVPR.2016.319