Understanding deep features with computer-generated imagery

Mathieu Aubry
UC Berkeley Université Paris-Est, LIGM (UMR CNRS 8049), ENPC
mathieu.aubry@imagine.enpc.fr

Bryan C. Russell
Adobe Research
brussell@adobe.com

Abstract

We introduce an approach for analyzing the variation of features generated by convolutional neural networks (CNNs) with respect to scene factors that occur in natural images. Such factors may include object style, 3D viewpoint, color, and scene lighting configuration. Our approach analyzes CNN feature responses corresponding to different scene factors by controlling for them via rendering using a large database of 3D CAD models. The rendered images are presented to a trained CNN and responses for different layers are studied with respect to the input scene factors. We perform a decomposition of the responses based on knowledge of the input scene factors and analyze the resulting components. In particular, we quantify their relative importance in the CNN responses and visualize them using principal component analysis. We show qualitative and quantitative results of our study on three CNNs trained on large image datasets: AlexNet [18], Places [43], and Oxford VGG [8]. We observe important differences across the networks and CNN layers for different scene factors and object categories. Finally, we demonstrate that our analysis based on computer-generated imagery translates to the network representation of natural images.

1. Introduction

The success of convolutional neural networks (CNNs) [18, 21] raises fundamental questions on how their learned representations encode variations in visual data. For example, how are different layers in a deep network influenced by different scene factors, the task for which the network was trained for, or the choice in network architecture? These questions are important as CNNs with different architectures and trained/fine tuned for different tasks have shown to perform differently [17, 43] or have different feature response characteristics [42]. An analysis of the features may help with understanding the tradeoffs across different trained networks and may inform the design of new architectures. It may also help the choice of CNN features for tasks where training or fine tuning a network is not possible, e.g. due to lack of labeled data.

Prior work has focused on a part-based analysis of the learned convolutional filters. Examples include associating filters with input image patches having maximal response [12], deconvolution starting from a given filter response [41], or by masking the input to recover the receptive field of a given filter [42] to generate "simplified images" [6, 34]. Such visualizations typically reveal the parts of an object [41] (e.g. "eye" of a cat) or scene [42] (e.g. "toilet" in bathroom). While these visualizations reveal the nature of learned filters, they largely ignore the question of the dependence of the CNN representation on continuous factors that may influence the depicted scene, such as 3D viewpoint, scene lighting configuration, and object style.

In this paper, we study systematically how different scene factors that arise in natural images are represented in a trained CNN. Example factors may include those intrinsic to an object or scene, such as category, style, and color, and extrinsic ones, such as 3D viewpoint and scene lighting configuration. Studying the variations associated with such factors is a nontrivial task as it requires (i) input data where the factors can be independently controlled and (ii) a procedure for detecting, visualizing, and quantifying each factor in a trained CNN.

To overcome the challenges associated with obtaining input data, we leverage computer-generated (CG) imagery to study trained CNNs. CG images offer several benefits. First, there are stores of 3D content online (e.g. Trimble 3D Warehouse), with ongoing efforts to curate and organize the data for research purposes (e.g. ModelNet [38]). Such data spans many different object categories and styles. Moreover, in generating CG images we have control over all rendering parameters, which allows us to systematically and densely sample images for any given factor. A database of natural images captured in controlled conditions and spanning different factors of variations, e.g. the NORB [22], ETH-80 [23] and RGB-D object [20] datasets, where different objects are rotated on a turntable and lighting is varied during image capture, are difficult and costly to collect. Moreover they do not offer the same variety of object styles present in 3D model collections, nor the flexibility given by rendering.

Given a set of rendered images generated by varying one or more factors, we analyze the responses of a layer for a trained CNN (e.g. "pool5" of AlexNet [18]). We perform a decomposition of the responses based on knowledge of the input scene factors, which allows us to quantify the relative importance of each factor in the representation. Moreover, we visualize the responses via principal component analysis (PCA).

Contributions. Our technical contribution is an analysis of contemporary CNNs trained on large-scale image datasets via computer generated images, particularly rendered from 3D models. From our study we observe:

- Features computed from image collections that vary along two different factors, such as style and viewpoint, can often be approximated by a linear combination of features corresponding to the factors, especially in the higher layers of a CNN.
- Sensitivity to viewpoint decreases progressively in the last layers of the CNNs. Moreover, the VGG fc7 layer appears to be less sensitive to viewpoint than AlexNet and Places.
- Relative to object style, color is more important for the Places CNN than for AlexNet and VGG. This difference is more pronounced for the background color than for foreground.
- The analysis we perform on deep features extracted from rendered views of 3D models is related to understanding their representation in natural images.

1.1. Related work

In addition to the prior work to visualize learned CNN filters [5, 11, 12, 32, 41, 42], there has been work to visualize hand-designed [37] and deep [25] features. Also related are recent work to understand the quantitative tradeoffs across different CNN layers for networks trained on large image databases [1, 40], designing CNN layers manually [7], and measuring equivariance and equivalence in CNNs [24].

Our use of a large CAD model dataset can be seen in the context of leveraging such data for computer vision tasks, e.g. object detection [2]. Contemporary approaches have used synthetic data with CNNs to render images for particular scene factors, e.g. style, pose, lighting [10, 19].

Our PCA feature analysis is related to prior work on studying visual embeddings. The classic Eigenfaces paper [36] performed PCA on faces. Later work studied nonlinear embeddings, such as LLE [29] and IsoMap [35]. Most related to us is the study of nonlinear CNN feature embeddings with the NORB dataset [14]. In contrast we study large-scale, contemporary CNNs trained on large image datasets.

Our multiple-factor study is related to approaches that learn to disentangle factors of variation. We note contemporary approaches for separating style and content via autoencoders [9] and learning to disentangle factors with a higher-order Boltzmann machine [28]. Finally, [4] observed faster mixing between modes and better generated samples in properly trained deep models on digits and face data due to a model's disentanglement of the underlying factors.

1.2. Overview

Our deep feature analysis begins by rendering a set of stimuli images by varying one or more scene factors. We present the stimuli images to a trained CNN as input and record the feature responses for a desired layer. Given the feature responses for the stimuli images we analyze the principal modes of variation in the feature space via PCA (section 2.1). When more than one factor is present we linearly decompose the feature space with respect to the factors and perform PCA on the feature decomposition (section 2.2). We give details of our experimental setup in section 3 and show qualitative and quantitative results over a variety of synthetic and natural images in section 4.

2. Approach for deep feature analysis

In this section we describe our approach for analyzing the image representation learned by a CNN. We seek to study how the higher levels of the CNN encodes the diversity present in a set of images. The minimal input for our analysis is a set of related images. We first describe our approach for analyzing jointly their features. What we can learn with such an approach is however limited since it cannot identify the origin of the variations of the input images. The factors of variation can be, e.g., variations in the style of an object, changes in its position, scaling, 3D rotation, lighting, or color. For this reason, we then focus on the case when the images are computer generated and we have full control of the different factors. In this case, we seek to separate the influence of the different factors on the representation, analyze them separately, and compare their relative importance.

2.1. Image collection analysis

We seek to characterize how a CNN encodes a collection of related images, Ω, e.g. images depicting a "car" or a black rectangle on white background. We sample images $r_\theta \in \Omega$ indexed by $\theta \in \Theta$. In the case of natural images Θ is an integer index set over the collection Ω. In the case of computer-generated images Θ is a set of parameters corresponding to a scene factor we wish to study (e.g. azimuth and elevation angles for 3D viewpoint, 3D model instances for object style, or position of the object in the image for 2D translation). Given a trained CNN, let $\tilde{F}^L(r_\theta)$ be a column vector of CNN responses for layer L (e.g. "pool5", "fc6", or "fc7" in AlexNet [18]) to the input image r_θ.

The CNN responses $\tilde{F}^L$ are high-dimensional feature vectors that represent the image information. However, since

Ω contains related images, we expect the features to be close to each other and their intrinsic dimension to be smaller than the actual feature dimension. For this reason, we use principal component analysis (PCA) [26] to identify a set of orthonormal basis vectors that capture the principal modes of variation of the features.

Given centered features $F^L(\theta) = \tilde{F}^L(r_\theta) - \frac{1}{|\Theta|}\sum_{t\in\Theta}\tilde{F}^L(r_t)$, where $|X|$ is the number of elements in set X, we compute the eigenvectors associated with the largest eigenvalues of the covariance matrix $\frac{1}{|\Theta|}\sum_{\theta\in\Theta}F^L(\theta)(F^L(\theta))^T$. The projection of the features onto the subspace defined by the D components with maximal eigenvalues corresponds to an optimal D-dimensional linear approximation of the features. We evaluate the intrinsic dimensionality of the features by computing the number of dimensions necessary to explain 95% of the variance. Moreover, we can visualize the embedding of the images by projecting onto the components with high variance.

2.2. Multiple factor analysis

In the case when we have control of the variation parameters, we can go further and attempt to decompose the features as a linear combination of uncorrelated components associated to the different factors of variation. Features decomposing linearly into different factors would be powerful and allow to perform image transformations directly in feature space and, e.g., to compare easily images taken under different viewpoints.

Let $\Theta_1,\ldots,\Theta_N$ be sets of parameters for N factors of variation we want to study. We consider an image of the scene with parameters $\theta = (\theta_1,\ldots,\theta_N)$, where $\theta \in \Theta = \Theta_1\times\cdots\times\Theta_N$. We assume the θ_k are sampled independently. We define marginal features $F_k^L(\theta_k)$ for scene factor k by marginalizing over the parameters for all factors except k:

$$F_k^L(t) = \mathbb{E}(F^L(\theta)|\theta_k = t) \quad (1)$$

$$= \frac{|\Theta_k|}{|\Theta|}\sum_{\theta\in\Theta|\theta_k=t}F^L(\theta). \quad (2)$$

Finally, we define a residual feature $\Delta^L(\theta)$, which is the difference of the centered CNN features $F^L(\theta)$ and the sum of all the marginal features $F_k^L(\theta_k)$. This results in the following decomposition:

$$F^L(\theta) = \sum_{k=1}^{N}F_k^L(\theta_k) + \Delta^L(\theta). \quad (3)$$

Using computer-generated images, we can easily compute this decomposition by rendering the images with all the rendering parameters corresponding to the sum in equation (2). Direct computation shows that all the terms in decomposition (3) have zero mean and are uncorrelated. This implies:

$$\text{var}(F^L) = \sum_{k=1}^{N}\text{var}(F_k^L) + \text{var}(\Delta^L). \quad (4)$$

We can thus decompose the variance of the features F^L as the sum of the variances associated to the different factors and a residual. When analyzing the decomposed features we report the relative variance $R_k^L = \text{var}(F_k^L)/\text{var}(F^L)$. We also report the relative variance of the residual $R_\Delta^L = \text{var}(\Delta^L)/\text{var}(F^L)$. A factor's relative variance provides an indication of how much the factor is represented in the CNN layer compared to the others. A high value indicates the factor is dominant in the layer and conversely a low value indicates the factor is largely negligible compared to the others. Moreover, a low value of the residual relative variance indicates the factors are largely separated in the layer. Note that $R_\Delta^L + \sum_{k=1}^{N}R_k^L = 1$ and the values of R_k^L and R_Δ^L do not depend on the relative sampling of the different factors. Also note that similar to section 2.1, we can study via PCA the principal modes of variation over the marginal features F_k^L for each factor k.

3. Experimental setup

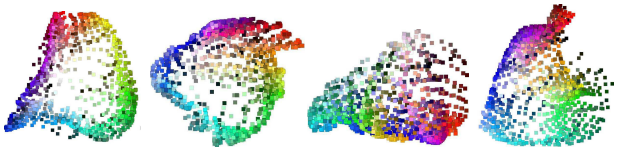
In this section we describe details of our experimental setup. In particular we describe the details of the CNN features we extract, our rendering pipeline, and the set of factors we seek to study.

3.1. CNN features

We study three trained CNN models: AlexNet [18], winner of the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [30], Places [43], which has the same architecture as AlexNet but trained on a large image database depicting scenes, and Oxford VGG [8] CNN-S network. In particular, we study the features of the higher layers "pool5", "fc6", and "fc7" of these networks. Note that the Oxford VGG architecture is different and the dimension of its "pool5" layer is two times larger than AlexNet and Places. We use the publicly-available CNN implementation of Caffe [16] to extract features for the different layers and pre-trained models for AlexNet, Places, and Oxford VGG from their model zoo.

3.2. Computer-generated imagery

We present two types of image stimuli as input: (i) 2D abstract stimuli consisting of constant color images or rectangular patches on constant background, which are described in detail in section 4.1, and (ii) rendered views from 3D models. For the latter we seek to render different object categories spanning many different styles from a variety of viewpoints and under different illumination conditions. We used as input CAD 3D models from the ModelNet database [38], which contains many models having different styles for a variety of



(a) AlexNet dims. 1, 2 (b) Places dims. 1, 2 (c) VGG dims. 1, 2 (d) VGG dims. 2, 3

Figure 1: PCA embeddings of constant-color images for the fc7 layer of different CNNs. The AlexNet and Places embeddings are similar to a hue color wheel, with more variation visible for the green and blue channels.

object classes. We downloaded the CAD models in Collada file format for the following object classes: chair (1261 models), car (485 models), sofa (701 models), toilet (191 models) and bed (258 models). We adapted the publicly-available OpenGL renderer from [3], which renders a textured CAD model with matte surfaces under fixed lighting configuration and allows the viewpoint to be specified by a 3×4 camera matrix [15]. We render the models under different lighting conditions and with different uniform colors.

We show results on two categories that have received the most attention in 3D-based image analysis: cars [13, 27], and chairs [2, 10]. Detailed results for three other categories are presented in the supplementary material and our quantitative results are averages over the 5 categories.

3.3. 3D scene factors

We study two types of factors affecting the appearance of a scene: (i) intrinsic factors – object category, style, and color, and (ii) extrinsic factors – 2D position, 2D scale, 3D viewpoint, and scene lighting configuration. The object category and style factors are specified by the CAD models from ModelNet. For object color we study grayscale matte surfaces (specified by Lambertian surface model), and constant-colored matte surfaces with the color uniformly sampled on a grid in RGB colorspace. For the 2D extrinsic factors, we uniformly sample 2D positions along a grid in the image plane and vary the 2D scale linearly. For 3D viewpoint we manually aligned all the 3D models to a canonical coordinate frame, i.e. all models are consistently oriented with respect to gravity and face the same direction, orbit the object at constant 3D distance and uniformly sample the azimuth and elevation angles with respect to the object's coordinate frame. Finally, we vary the scene lighting such that the source light varies from left to right and front to back on a uniform grid.

For the quantitative experiments we sampled 36 azimuth angles (keeping the elevation fixed at 10 degrees) for rotation, 36 positions for translation, 40 scales, 36 light positions, and 125 colors. For the visualizations we sampled 120 azimuth angles, 121 light positions, and 400 positions to make the embeddings easier to interpret. We checked that the different sampling did not change our quantitative results, which was expected since our method is not sensitive to the relative number of samples for each factor.

Table 1: Relative variance of the aspect ratio, 2D position, and residual feature for our synthetic rectangle experiment with AlexNet. Notice that the relative variance of the aspect ratio increases with the higher layers while 2D position decreases, which indicates that the features focus more on the shape and less on the 2D location in the image.

	2D position	Aspect ratio	Δ^L
AlexNet, pool5	49.8 %	9.5 %	40.8 %
AlexNet, fc6	45.1 %	22.3 %	32.6 %
AlexNet, fc7	33.9 %	37.0 %	29.1 %

4. Results

In this section we highlight a few results from our experiments on CNN feature analysis. The supplementary material reports our detailed quantitative results for all object categories and provides a visualization tool to interactively select and compare embeddings for the first ten PCA components.

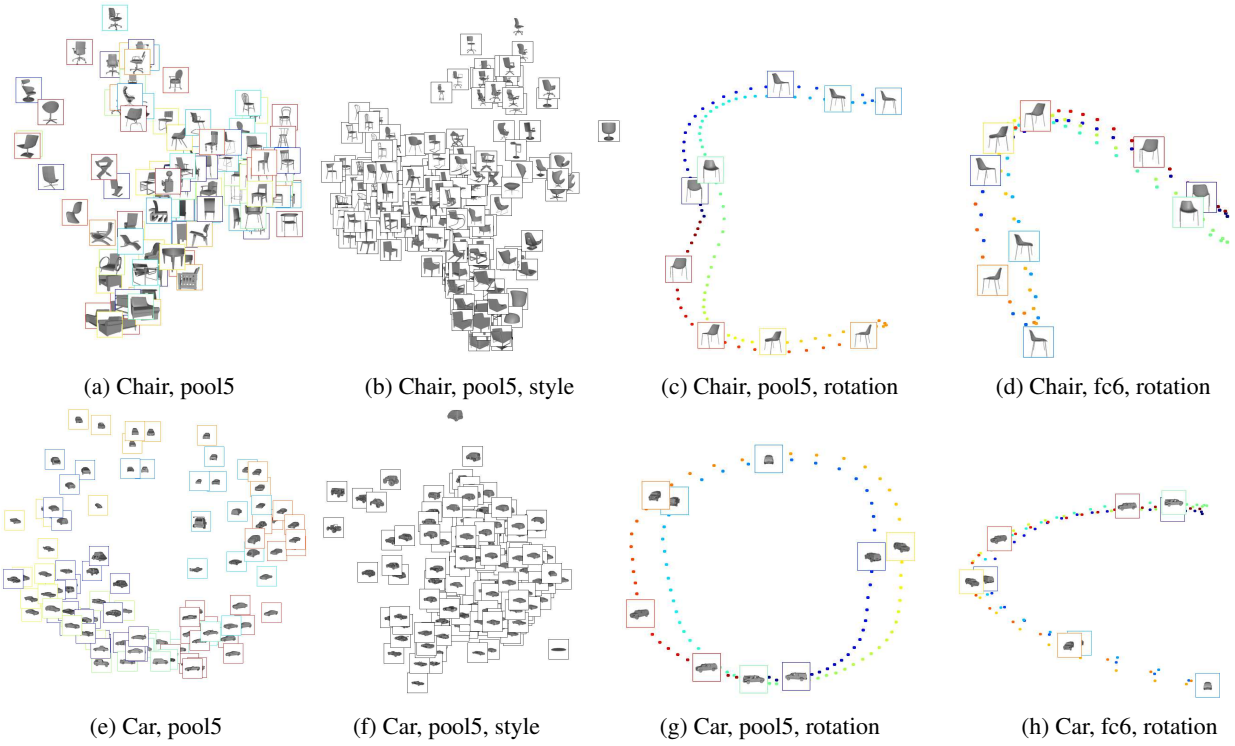
We first report results for manually-designed 2D stimuli in section 4.1 and then for rendered views of 3D models from several object categories in section 4.2. We finally show in section 4.3 that our results obtained with computer-generated images are related to natural images.

4.1. 2D abstract stimuli

In this section we apply the deep feature analysis of section 2 on manually-designed 2D abstract stimuli presented to a trained CNN. We first perform a PCA analysis on a single factor, color. Next, we perform two-factor quantitative analyses on the aspect ratio/2D position of a rectangle.

Uniform color. We perform PCA on a set of images with constant color, as described in section 2.1. We sampled 1331 colors uniformly on a grid in RGB color space. The resulting embedding for the fc7 layers of the three CNNs are shown in figure 1. The resulting embeddings for the AlexNet and Places CNNs are surprisingly similar to a hue color wheel, with more variation visible for blues and greens and less for reds and violets. VGG has a different behavior, with the first dimension similar to saturation. The embedding corresponding to the second and third dimensions is similar to that of the other CNNs. The number of PCA dimensions necessary to explain 95% of the variance is approximately 20 for all three networks and all three layers, which is higher than the three dimensions of the input data.

Position and aspect ratio. We used the methodology of section 2.2 to study the features representing a small black



(a) Chair, pool5 (b) Chair, pool5, style (c) Chair, pool5, rotation (d) Chair, fc6, rotation

(e) Car, pool5 (f) Car, pool5, style (g) Car, pool5, rotation (h) Car, fc6, rotation

Figure 2: **Best viewed in the electronic version.** PCA embeddings (dims. 1,2) of AlexNet features for "chairs" (first row) and "cars" (second row). Column 1 – Direct embedding of the rendered images without viewpoint-style separation. Columns 2,3 – Embeddings associated with style (for all rotations) and rotation (for all styles). Column 4 – Rotation embedding for fc6, which is qualitatively different than pool5. Colors correspond to orientation and can be interpreted via the example images in columns 3,4. Similar results for other categories and PCA dimensions are available in the supplementary material.

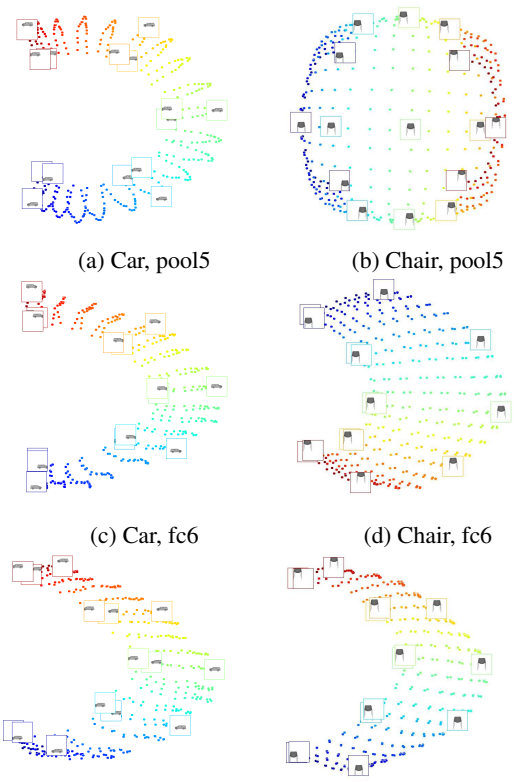
rectangle located at different positions with different aspect ratios on a white background. The rectangle area was kept constant at 0.26^2 of the image area. We consider the position and aspect ratio as two factors of variation and sample 36 positions on a grid and 12 aspect ratios on a log scale. The variance explained by each factor for the different layers of AlexNet is presented in table 1. For all three networks the relative variance associated to the position decreases, which quantitatively supports the idea that the higher layers have more translation invariance. In contrast the relative variance associated to the aspect ratio increases for the higher CNN layers. For AlexNet, less than 10% of the relative variance for pool5 is explained by the aspect ratio alone, while it explains 37% of the relative variance for fc7. Also, the relative variance associated with the residual decreases for the higher CNN layers, which indicates the two factors are more easily linearly separated in the higher layers.

Remarks. As the CNNs were not trained on the 2D artificial stimuli presented in this section, we find it somewhat surprising that the embeddings resulting from the above feature analysis is meaningful. From our experiments we saw that the CNNs learn a rich representation of colors, identifying in particular variations similar to hue and saturation. Moreover, the last layers of the network better encode translation invariance, focusing on shape. These results will be confirmed and generalized on more realistic stimuli in the next sections.

4.2. Object categories

In this section we want to explore the embedding generated by the networks for image sets and factors related to the tasks for which they are trained, namely object category classification in the case of AlexNet and VGG. We also compare against the CNN trained on Places and show that our analysis can apply to complete complex networks, such as GoogLeNet [33]. We thus select an object category and, using rendered views of 3D models, we analyze how the CNN features are influenced by the style of the specific instances as well as different transformations and rendering parameters. The parameter sampling for each experiment is described in section 3.3.

Model–orientation separation. The first variation we study jointly with style is the rotation of the 3D model. The first column of figure 2 visualizes the PCA embedding of the



(a) Car, pool5 (b) Chair, pool5

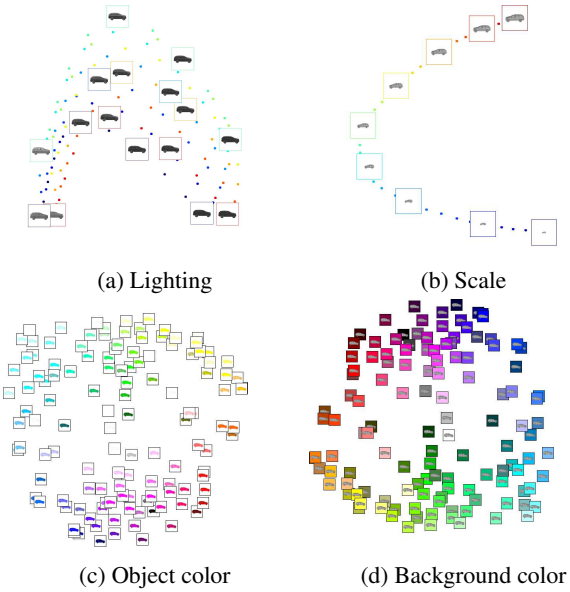
(c) Car, fc6 (d) Chair, fc6

(e) Car, fc7 (f) Chair, fc7

Figure 3: PCA embeddings for 2D position on AlexNet.

resulting pool5 features. This embedding is hard to interpret because it mixes information about viewpoint (important for cars) and instance style (important for chairs). To separate this information, we perform the decomposition presented in section 2. The decomposition provides us with embedding spaces for style and viewpoint and associates to each model and viewpoint its own descriptor. We visualize the embeddings in figure 2; the second column corresponds to style and the third to viewpoint. Note that the different geometries of the two categories lead to different embeddings of rotation in pool5. While a left-facing car typically looks similar to a right-facing car and is close in the feature space (figure 2g), a right-facing chair is usually different from left-facing chairs and is far in the embedding (figure 2c). The last column shows the viewpoint embedding for fc6. The comparison of the last two columns indicates that much viewpoint information is lost between pool5 and fc6 and that fc6 is largely left-right flip invariant. A potential interesting future direction could be to interpret the viewpoint embeddings relative to classic work on mental rotation [31].

Translation, scale, lighting, color. We repeated the same experiment for the following factors: 2D translation, scale, light direction, background color, and object color. For



(a) Lighting (b) Scale

(c) Object color (d) Background color

Figure 4: PCA embeddings for different factors using AlexNet pool5 features on "car" images. Colors in (a) correspond to location of the light source (green – center).

simplicity and computational efficiency, we considered in all experiments a frontal view of all the instances of the objects. The framework allows the same analysis using the object orientation as an additional factor. The embeddings associated with AlexNet features for translation of cars and chairs are shown in figure 3. Note that similar to rotations, the embedding corresponding to cars and chairs are different, and that the first two components of the fc6 features indicate a left-right flip-invariant representation. The embeddings for the pool5 layer of the car category for the other factors are shown figure 4.

Quantitative analysis: viewpoint. We analyze the relative variance explained by the 3D rotation, translation, and scale experiments. While the variance was different for each factor and category, the variation across the layers and networks was consistent in all cases. For this reason we report in table 2 an average of the variance across all five categories and all three factors. We refer the reader to the supplementary material for detailed results. The analysis of table 2 reveals several observations. First, the proportion of the variance of deeper layers corresponding to viewpoint information is less important, while the proportion corresponding to style is more important. This corresponds to the intuition that higher layers are more invariant to viewpoint. We also note that the residual feature Δ^L is less important in higher layers, indicating style and viewpoint are more easily separable in those layers. These observations are consistent with our results of section 4.1. Second, the part of the variance associated with style is more important in the

Table 2: Relative variance and intrinsic dimensionality averaged over experiments for different object categories and viewpoints (3D orientation, translation, and scale). Each cell: top – rel. variance; bottom – intrinsic dim. We do not report the intrinsic dim. of Δ^L since it is typically larger than 1K across the experiments and expensive to compute.

		pool5	fc6	fc7
Viewpoint	Places	26.8 % 8.5	21.4 % 7.0	17.8 % 5.9
	AlexNet	26.4 % 8.3	19.4 % 7.2	15.6 % 6.0
	VGG	21.2 % 10.0	16.4 % 7.7	12.3 % 6.2
Style	Places	26.8 % 136.3	39.1 % 105.5	49.4 % 54.6
	AlexNet	28.2 % 121.1	40.3 % 125.5	49.4 % 96.7
	VGG	26.4 % 181.9	44.3 % 136.3	56.2 % 94.2
Δ^L	Places	46.8 %	39.5 %	32.9 %
	AlexNet	45.0 %	40.3 %	35.0 %
	VGG	52.4 %	39.3 %	31.5 %

fc7 layer for VGG than in AlexNet and Places. Also, the part associated with the viewpoint and residual is smaller. Note that this does not hold in pool5, where the residual is important for the VGG network. This effect may be related to the difference in the real and intrinsic dimension of the features. The intrinsic dimension of the style component of VGG pool5 features is larger and decreases from pool5 to fc7. On the contrary, the intrinsic dimensionality of AlexNet has smaller variation across layers. Finally, we note that the intrinsic dimensionality of the fc7 style feature of Places is smaller than the other networks. This may indicate that it is less rich, and may be related to the fact that identifying the style of an object is less crucial for scene classification. We believe it would be an interesting direction for future work to study how the improved performance of VGG for object classification is related to the observed reduced sensitivity to viewpoint. Figure 5 shows relative variance of the residual feature on the entire GoogLeNet [33] network. Notice that the residual progressively decreases in the higher network layers. Also, the different parts of the inception modules behave differently. We believe that understanding the role of different parts of a network is key to better architecture design. The supplementary material will provide similar visualizations for all networks and factors.

Quantitative analysis: color. We report in table 3 the average across categories of our quantitative study for object and background color. The results are different from those of

Table 3: Average relative variance over five classes for color/style separation.

Foreground/Style		pool5	fc6	fc7
FG color	Places	23.4 %	29.4 %	34.9 %
	AlexNet	23.2 %	24.0 %	24.0 %
	VGG	15.0 %	22.6 %	25.0 %
Style	Places	48.9 %	41.3 %	40.3 %
	AlexNet	56.6 %	52.1 %	52.5 %
	VGG	59.0 %	51.4 %	51.3 %
Δ^L	Places	27.7 %	29.3 %	24.8 %
	AlexNet	20.3 %	24.0 %	23.5 %
	VGG	26.0 %	25.9 %	23.6 %

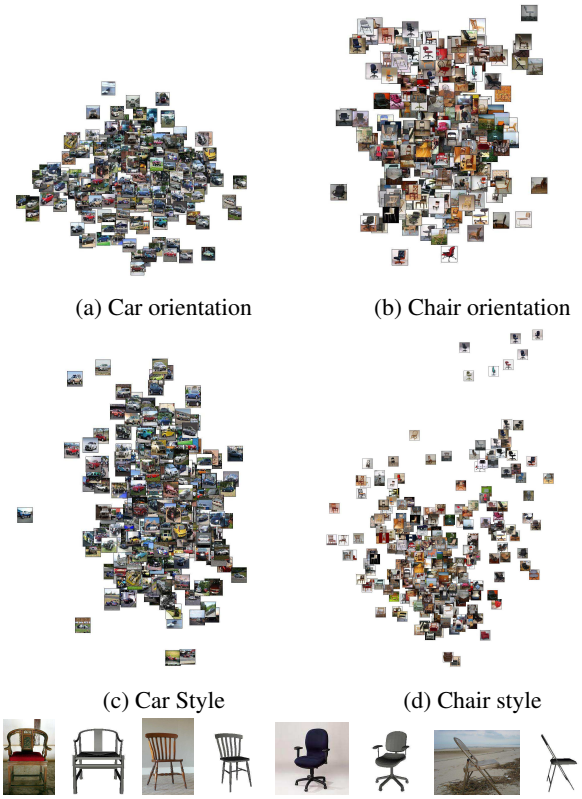
Background/Style		pool5	fc6	fc7
BG color	Places	24.3 %	29.6 %	35.1 %
	AlexNet	17.3 %	16.2 %	14.4 %
	VGG	9.1 %	13.8 %	14.3 %
Style	Places	51.5 %	40.7 %	40.9 %
	AlexNet	63.7 %	59.4 %	61.8 %
	VGG	71.4 %	64.3 %	65.3 %
Δ^L	Places	24.2 %	29.7 %	24.0 %
	AlexNet	19.0 %	24.4 %	23.9 %
	VGG	19.5 %	22.0 %	20.4 %

viewpoint. First, we observe that a larger part of the variance of the features of the Places network is explained by the color in all layers. This may be related to the fact that color is a stronger indicator of the scene type than it is of an object category. Second, while the part of the variance explained by foreground and background color is similar in the fc7 feature of the Places network, it is much larger for the foreground object than for the background object in AlexNet and VGG. Once again, one can hypothesize that it is related to the fact that the color of an object is more informative than the color of its background for object classification. Finally, we note that similarly to our previous experiments, the difference between networks is present in pool5 and increases in the higher layers, indicating that the features become more tuned to the target task in the higher layers of the networks.

4.3. Natural images

Embedding. We used ImageNet [30] images to study the embeddings of natural images. Since we have no control over the image content, we cannot perform a detailed analysis of the different factors similar to the previous sections. Our only choice is to consider the images altogether. The direct embedding of natural images is possible but hard to interpret. We can however project the images in the spaces discovered in section 4.2. The resulting embeddings for style and viewpoint are shown in figure 6 and are similar to the embeddings obtained with the CAD models.

Figure 5: **Best viewed in the electronic version.** Residuals of the viewpoint-style separation experiments for GoogLeNet [33]. Warmer colors denote higher relative variance. The network tends to increasingly separate the factors in the higher layers. Also, the layers inside each inception module behave differently. E.g., until the end of the network, the information is not well separated in the 1x1 convolution, while it is always better separated in the 5x5 convolution.



(a) Car orientation (b) Chair orientation

(c) Car Style (d) Chair style

(e) 2D-3D retrieval examples

Figure 6: PCA embeddings over AlexNet pool5 features for cars and chairs with orientation and style separated.

2D-3D instance recognition. The observed similarity of the embeddings for natural and rendered images motivates an application to retrieve a 3D model of an object present in an image without explicitly performing alignment or detection. The approach is different from approaches in the 2D-3D matching community that find explicit correspondences between the models and the image. We tested this idea using the chair category and computing similarity as dot product between AlexNet features. We rendered 36 azimuth and 4 elevation angles to span typical viewpoints depicted in the natural images. To improve efficiency, we reduced the dimension of our features to 1000 via PCA. This allows us to perform nearest-neighbor retrieval between 1000 natural images and the 144 rendered views of 1261 3D models in approximately 22 seconds total using a Python implementation with pre-computed features. We visualize results in figure 6e and in the supplementary material. We evaluated the viewpoint accuracy using the annotations of [39] and found that the orientation error was below 20 degrees for 60% of the images using pool5 features, 39% using fc6, and 26% using fc7. This is consistent with our earlier finding that orientation is not as well represented in the higher layers. We conducted a user study on Mechanical Turk to evaluate the quality of the style matching for the images where the orientation was correctly estimated with the pool5 features. The workers were presented with a pair of images and asked to judge if the style of the chairs and their orientation were similar or different, similar to [2]. Each pair was evaluated 5 times. There was agreement in 75% of the cases that the match was fair and in 18% that it was exact. While the above results could probably be beaten by state-of-the-art 2D-3D matching techniques or simply by adding position and scale to our database, they show that our analysis of rendered 3D models is pertinent for understanding the CNN representation of natural images.

5. Conclusion

We have introduced a method to qualitatively and quantitatively analyze deep features by varying the network stimuli according to factors of interest. Utilizing large collections of 3D models, we applied this method to compare the relative importance of different factors and have highlighted the difference in sensitivity between the networks and layers to object style, viewpoint and color. We believe our analysis gives new intuitions and opens new perspectives for the design and use of convolutional neural networks.

Acknowledgments. We are grateful to Aaron Hertzmann and Alexei Efros for insightful discussions and feedback. Mathieu Aubry was partly supported by ANR project Semapolis ANR-13-CORD-0003, Intel, a gift from Adobe, and hardware donation from Nvidia.

References

[1] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 2014.

[2] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *CVPR*, 2014.

[3] M. Aubry, B. C. Russell, and J. Sivic. Painting-to-3D model alignment via discriminative visual elements. *ACM Transactions on Graphics*, 33(2), 2014.

[4] Y. Bengio, G. Mesnil, Y. Dauphin, and S. Rifai. Better mixing via deep representations. In *ICML*, 2013.

[5] P. Berkes and L. Wiskott. On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural Computation*, 2006.

[6] I. Biederman. *Visual object recognition*, volume 2. MIT press Cambridge, 1995.

[7] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE PAMI*, 35(8):1872–1886, 2013.

[8] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC.*, 2014.

[9] B. Cheung, J. Livezey, A. Bansal, and B. Olshausen. Discovering hidden factors of variation in deep networks. In *ICLR workshop*, 2015.

[10] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *CVPR*, 2015.

[11] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical report, University of Montreal, 2009.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[13] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[14] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[15] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[16] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[17] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemöller. Recognizing image style. In *Proc. BMVC.*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. *arXiv preprint arXiv:1503.03167*, 2015.

[20] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, 2011.

[21] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, 1989.

[22] Y. LeCun, F.-J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004.

[23] B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *CVPR*, volume 2, pages II–409. IEEE, 2003.

[24] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *CVPR*, 2015.

[25] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, 2015.

[26] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(11):559–572, 1901.

[27] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *CVPR*, 2012.

[28] S. Reed, K. Sohn, Y. Zhang, and H. Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML*, 2014.

[29] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575*, 2014.

[31] R. Shepard and J. Metzler. Mental rotation of three dimensional objects. *Science*, 171(972):701–3, 1971.

[32] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR workshop*, 2014.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[34] K. Tanaka. Neuronal mechanisms of object recognition. *Science*, 262(5134):685–688, 1993.

[35] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(550):2319–2323, December 2000.

[36] M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neuroscience*, 3(1):71–86, 1991.

[37] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing object detection features. In *ICCV*, 2013.

[38] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A deep representation for volumetric shape modeling. In *CVPR*, 2015.

[39] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *WACV*, 2014.

[40] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

[41] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *ICLR*, 2015.

[43] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using Places database. In *NIPS*, 2014.