Maximilian Eibl, Martin Gaedke. (Hrsg.): INFORMATIK 2017,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 907

Deep Convolutional Neural Networks for Pose Estimation in Image-Graphics Search

Markus Eberts[1], Adrian Ulges[2]

Abstract: Deep Convolutional Neural Networks (CNNs) have recently been highly successful in various image understanding tasks, ranging from object category recognition over image classification to scene segmentation. We employ CNNs for pose estimation in a cross-modal retrieval system, which - given a photo of an object - allows users to retrieve the best match from a repository of 3D models. As our system is supposed to display retrieved 3D models from the same perspective as the query image (potentially with virtual objects blended over), the pose of the object relative to the camera needs to be estimated. To do so, we study two CNN models. The first is based on *end-to-end learning*, i.e. a regression neural network directly estimates the pose. The second uses *transfer learning* with a very deep CNN pre-trained on a large-scale image collection. In quantitative experiments on a set of 3D models and real-world photos of chairs, we compare both models and show that while the end-to-end learning approach performs well on the domain it was trained on (graphics) it suffers from the capability to generalize to a new domain (photos). The *transfer learning* approach on the other hand handles this domain drift much better, resulting in an average angle deviation from the ground truth angle of about 14 degrees on photos.

Keywords: pose estimation; image retrieval; deep learning; transfer learning

1 Introduction

Deep Learning – a major current trend in machine learning research – refers to the application of neural networks with multiple layers. Such networks have been particularly successful in various image understandings tasks, be it object category recognition [Ru15], gaze estimation [Zh15] or the segmentation of road scenes [Ho16]. A key concept in the area are convolutional neural networks (CNNs), which learn filters that serve as local feature detectors strongly adapted to the task at hand. Furthermore, by "stacking" multiple layers, i.e. feeding the filter responses from one layer to subsequent filters, feature detectors of increasing complexity develop [ZF13].

In this paper, we apply CNNs for cross-modal image-graphics retrieval. Given is a repository of 3D models of objects (say, pieces of furniture or other products of interest). A user takes

[1] RheinMain University of Applied Sciences, DCSM Department, Unter den Eichen 5, D-65195 Wiesbaden / Germany, markus.eberts@hs-rm.de
[2] RheinMain University of Applied Sciences, DCSM Department, Unter den Eichen 5, D-65195 Wiesbaden / Germany, adrian.ulges@hs-rm.de

doi:10.18420/in2017_92

a photo of a real-world object, marks the object in the photo, and our system automatically retrieves the best-fitting 3D model from the repository. Our system has direct applications in smart shopping: In case the user finds a suitable 3D model, it can be modified to the user's individual needs, leading to a fully customized product. Other applications include virtual furnishing (say, replacing a piece of furniture in a photo with a different one). More generally, our matching might form a cornerstone for smart applications offering multi-modal interactions involving 3D graphics (be it products, pieces of art, buildings, or animated scenes in gaming).

To realize image-graphics retrieval, we follow a view-based approach: A CNN learns to estimate from rendered training views *which* object is visible in the target image, and from what *perspective*. In this paper, we focus on the latter question (commonly referred to as *pose estimation*): Given a photo of an object (say, a chair), can a CNN estimate from what perspective the object was filmed? A particular challenge here is that acquiring training data with pose ground truth is cost-intensive. As a proxy, we train our CNN on views rendered from 3D models, which can be acquired (including ground truth) at a large scale (see Figure 1). Note, however, that there is a *domain drift* between graphics and camera-captured photos: Lighting conditions in low-quality graphics are simplified, capture effects such as camera noise and blur are usually not taken into account. Beyond this, artificial backgrounds differ from real ones, and 3D training models may not exactly fit the object in the photo.

There are two general strategies to deal with this challenge: First, a – usually rather small-scale model – CNN can be trained from scratch for the target problem at hand (here, on rendered content) [Zh15, Gi16]. Second, larger-scale networks (see [Sz15, KSH12, He15]) – which are pre-trained on large-scale image collections and show remarkable accuracy in recognizing large numbers of diverse object categories – can be *transferred* to novel problems [Yo14]. While the former network (called **from-scratch** in the following) may be better adapted to the task at hand, the latter one (referred to as **transfer**) may provide better robustness with respect to domain drift, as training has happened on real-world photos. The key contribution of our work is an experimental comparison of both approaches on a dataset of 200 3D models of chairs (about 40.000 rendered views) and 263 photos with pose ground truth, in which we demonstrate that robustness to domain drift is crucial, such that the **transfer** approach is preferable.

2 Related Work

CNNs are considered state-of-the-art in various image understanding tasks [Ru15, Zh15, Ho16, Gi16], and have also been trained on rendered content before: For example, Hueting et al.'s image-graphics matching includes features pulled from a CNN [HOM15]. Often, transfer learning is applied, i.e. pre-trained CNNs are (after an optional adaption process) applied to different domains. Zamir et al. [Za16] demonstrate that a network trained on 3D proxy tasks (here, wide-baseline matching) can be applied to other related tasks, even without fine-tuning. Yosinski et al. [Yo14] demonstrate strong transferability of features learned

Fig. 1: Training views rendered with *Flickr skybox* (left) or plain white background (right).

on the large-scale ImageNet dataset [Ru15]. We follow this line of work (and will study a network pre-trained on the same dataset), but focus on pose estimation. Domain drift when transferring from graphics to photos has been reported by Hofmann et al. [Ho16]. They suggested an adversarial training, which enforces the network to produce signals of similar distribution for both graphics and real-world photos. Similarly, Long and Wang [LW15] achieve an adaptation between domains by minimizing the domain discrepancy of a CNN's inner layers' signals.

As an alternative to CNNs, hand-crafted features based on gradient statistics such as HOG [DT05] or SIFT [Lo04] have been widely used both for object recognition and pose estimation. Aubrey et al. [Au14] match photos of chairs with a large-scale repository of 3D models. They carefully select local HOG-based features by discriminative training and combine them using a star model. Brachmann et al. [Br16] employ random decision forests predicting posteriors for objects and object parts, which are processed by a RANSAC-based hypothesis testing. Finally, in their *spatial pyramid* model, Lazebnik et al. [LSP06] use windowed histograms over vector-quantized SIFT features (dubbed *visual words*), encoding which types of SIFT features occur in which subregions of the image (we include their approach as a baseline in our experiments).

3 Approach

Our approach renders 3D models into sample views, which serve as ground truth for training a CNN-based pose estimator, using either a *from-scratch* or a *transfer* network:

Rendering For each 3D model, we render training views by utilizing the Python API of Blender, an open source 3D computer graphics software. Our system scales all models to the same size and shifts their center of mass to the coordinate system's origin. The camera points at the origin (the center of the model) with a roll angle of zero. We use a subdivision approach to sample camera positions in regular distance[3]. The background is either plain white or an enclosing skybox textured with random Flickr images. As we assume the user to select the object of interest in our UI, we crop all training images to the object's bounding box (optionally, the bounding box is extended to be of squared shape).

[3] http://www.gamedev.sk/triangulation-of-a-sphere

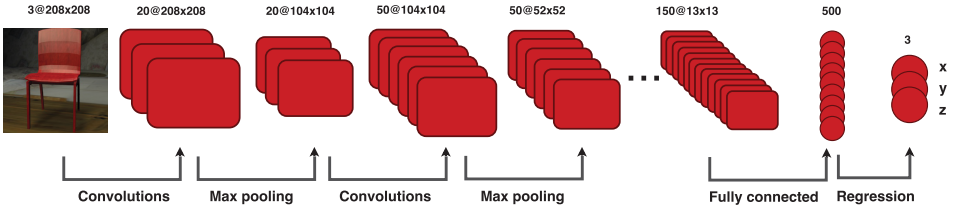

Fig. 2: Architecture of the from-scratch convolutional neural network.

Approach 1: From-scratch Our main contribution is the comparison of different CNN-based learners for pose estimation. Our first approach employs a small-scale CNN trained from scratch on rendered graphics. It applies *end-to-end learning*, i.e. it directly predicts the object pose and is trained by minimizing the angular error of pose estimation. The network is layed out in Figure 2 and follows common practice with CNN-based models: The input image is filtered with convolution masks, which serve as local feature detectors. These convolution operations are stacked in 4 layers. In each layer, an increasing number of filters (20, 50, 100, 150) of mask size 5x5 is applied. We use rectified linear units (ReLU) as activation functions. Each convolutional layer is also followed by a max-pooling, which scales down feature maps and limits the dimensionality of intermediate signals, resulting in (208x208, 104x104, 52x52, 26x26, 13x13) feature maps. In the end, two fully connected layers process the filter responses to estimate from which perspective the target object is filmed. The final layer performs plain linear regression and outputs the 3-dimensional camera position. As our end user crops the query image to the target object and the cropped region is rescaled to a standard size, we normalize the distance between the camera and object and estimate only camera positions on the 3D unit sphere. We define the loss function L for training by calculating the squared Euclidean distance between the camera positions $\mathbf{c}$ (estimated by the network) and $\mathbf{c}'$ (ground truth), both normalized to a Euclidean norm of one:

$$L(\mathbf{c}, \mathbf{c}') := (\mathbf{c}_x - \mathbf{c}'_x)^2 + (\mathbf{c}_y - \mathbf{c}'_y)^2 + (\mathbf{c}_z - \mathbf{c}'_z)^2$$

Approach 2: Transfer The second approach utilizes the well-known Inception-v3 network [Sz15], a very deep CNN pre-trained on the large-scale ImageNet dataset for recognizing 1000 object categories. The Inception-v3 network consists of ten so-called inception layers. Each contains several convolution and max pooling operations, and outputs multiple feature maps. Basically, the internal signals of the network $h_1(\mathbf{x})$, $h_2(\mathbf{x})$, ..., $h_{10}(\mathbf{x})$ indicate how well the different regions of the input image $\mathbf{x}$ "fit" a particular layer's detectors. Thereby, feature maps pulled from the inception network are rescaled to 8×8 pixels, which we found to work best in earlier experiments.

As has been demonstrated before [Yo14], the tensors $h_i(\mathbf{x})$ can be fed as feature vectors into subsequent learners. We follow this approach using a 2-Nearest-Neighbor matching: Given an image, we estimate the two training views with the most similar inception-based features

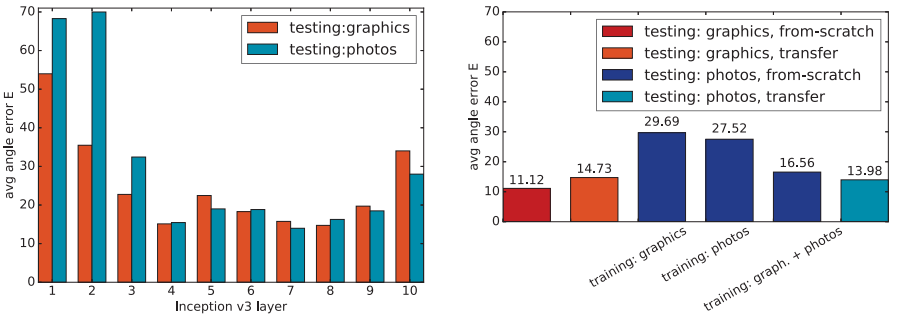

Fig. 3: Average angle error of the from-scratch and transfer CNN (left plot). The test was conducted on either graphics (left side of plot) or photos (right side of plot). Comparison of Inception-v3 layers and their effects on the average angle error (right plot). Tested on graphics (orange) and photos (teal).

and average their camera positions to obtain our pose estimate. Note that we do not perform any refinement of the Inception-v3 network on the target domain (pose estimation),

4 Experiments

We evaluate both CNN based approaches (from-scratch vs. transfer) and also include a baseline experiment that utilizes SIFT features. Thereby, we tackle several questions: (1) How well do the approaches generalize from rendered graphics to photos? (2) Are the features produced by the deep Inception CNN, which is pre-trained on a huge collection of photos, sufficient for pose estimation? (3) Which layers of the Inception CNN prove most useful for the kNN-based pose estimation? Our dataset consists of 200 chair 3D models, which we filtered from a larger-scale dataset of Aubrey at al.'s [Au14] by removing models with broken geometry. We tested all approaches on both graphics data and photos (details follow in the upcoming subsections). As an error measure, we use the angle between the estimated camera position $\mathbf{c}$ and the ground truth position $\mathbf{c}'$:

$$E(\mathbf{c}, \mathbf{c}') := \arccos\left(\frac{\mathbf{c}^\top \cdot \mathbf{c}'}{||\mathbf{c}|| \cdot ||\mathbf{c}'||}\right)$$

Transfer: Inception Layers The transfer approach relies on features pulled from an inner layer of the Inception-v3 network. We first examine which layer $h_i(\mathbf{x})$ to pull features from. Figure 3 (left) shows the angle error plotted over ($i = 1, ..., 10$), both when testing on graphics content and on photos (details will follow in the next section). We observe that higher levels of the Inception model tend to overadapt to the specific data the network was initially trained for, while lower layers lack expressive power (this is in line with earlier findings by Yosinski et al. [Yo14]). Intermediate layers seem to offer a good compromise,

such that we will use Layer 7 in the remaining experiments. The feature vectors $h_7(\mathbf{x})$ consist of 768 filter responses of shape 8×8 (about 50.000 dimensions).

Testing on Graphics We first evaluate both CNN approaches on graphics material. For the **from-scratch** approach, we split the models into 150 training chairs and 50 test chairs (cross-validation was omitted due to time constraints). Each training chair was rendered 400 times with random poses and Flickr backgrounds. All input to the CNN was cropped to a square region. We use a batch size of 50, a learning rate of 0.01 and applied 50% dropout. In the case of **transfer learning**, we performed leave-one-out cross-validation on all 200 models. We generated 209 training views per model by subdivision sampling of the upper hemisphere, rendered with plain white background. Sampling multiple backgrounds per view was omitted due to time and memory constraints (nearest neighbor indexing on 50K-dimensional non-sparse vectors proved resource-intensive). All input was cropped to the object bounding box. The graphics test views were rendered with a Flickr skybox background and a different number of subdivision steps, resulting in views that deviate in about 7 degrees from the transfer training views.

Results of the experiments are illustrated in Figure 3 (right, reddish). End-to-end learning (with an average angle error of about 11°) outperforms the Inception network (15°). Obviously, the CNN trained directly on the target domain (graphics content, pose estimation) is better suited compared to the Inception signals.

Testing on Photos We also tested both CNN approaches with real photos taken at a local furniture store. We captured a dataset of 263 photos of 40 different chairs with varying backgrounds, clutter and lightning conditions (note that shapes and colors tended to differ strongly from the 200 3D reference models). To acquire ground truth camera poses, each picture was shot twice from the same perspective, with one of the shots containing a chessboard marker placed on the seat of the chair. Based on the marker, we calculate the ground truth pose. Figure 3 (right, blueish) illustrates the results of the experiment. The **from-scratch** network trained on graphics performed much worse (angle error of 29°, "training: graphics") when generalizing to photos, due to the aforementioned domain drift. To investigate how the from-scratch network would adapt to the target domain, we performed two more control experiments: (1) "training: photos"where the CNN was trained on roughly 60% of the photos and tested on the remaining 40%. This gave a comparable angle error of 27°. Here, we observed strong overfitting, as the training set was quite limited. (2) "training: graphics+photos", where the CNN was pre-trained on graphics and then refined on the 60% training photos. Here, we observed a strong improvement with an angle error of 16°, indicating that even limited labeled data on the target domain can be extremely useful. Finally, the **transfer** approach gave the best performance, outperforming all runs of the from-scratch network with an average angle error of about 14°. This demonstrates the astonishing capability of transfer learning, and shows that training on photo material is extremely useful.

Baseline: Spatial Pyramids with hand-crafted Features Finally, we also tested a baseline

Fig. 4: Pose estimation results on photos. Views estimated by the from-scratch CNN (left), with the left sample showing an angle error of about 30° (in line with the avg. angle error without refinement) and the second to left sample an angle error of 16° (in line with the avg. angle error with refinement). Nearest neighbors estimated by transfer learning (right), with the fourth example showing an angle error of about 13.5° (the avg. angle error achieved with transfer learning is 13.98°).

using hand-crafted features. The approach is based on the well-known SIFT descriptors [Lo04] for local feature extraction. The features produced by SIFT are invariant with respect to translation, rotation and scale, and also robust with respect to image characteristics like noise and changes of illumination. We detect interest points using dense sampling (every 8 pixels), vector-quantizing the SIFTs to 100 clusters, and generate normalized windowed histogram corresponding to the *spatial pyramid* scheme by Lazebnik et al. [LSP06]. To estimate pose, we find the most similar training view to the query image by a nearest neighbor search over the resulting histograms (using the histogram intersection kernel). Even with tuning of the parameters, this approach resulted in an average angle error of 52.10° and is significantly outperformed by all our CNN models. We observed that nearest neighbor views found by the approach were often the most strongly textured ones.

5 Conclusions

We have studied deep CNNs for pose estimation in image-graphics retrieval. Our key finding has been that a very deep CNN pre-trained on the ImageNet dataset offers remarkable robustness when generalizing between domains (graphics vs. photos) and tasks (recognition vs. pose estimation). Our results are in line with current research [Za16] indicating that deep CNNs have the potential to generalize between diverse image understanding tasks. This might ultimately result in general-purpose vision systems, which would form a corner stone for image understanding in various vision-based smart systems.
Interesting directions for our future research persist in (1) a fine-tuning of the Inception-v3 network to the target domain (note that we applied the network off-the-shelf so far), and (2) a fine-grain refinement of angle estimates (our current k-NN solution suffers from quantization effects in the reference views).

References

[Au14] Aubry, Mathieu; Maturana, Daniel; Efros, Alexei; Russell, Bryan; Sivic, Josef: Seeing 3D chairs: Exemplar Part-based 2D-3D Alignment using a Large Dataset of CAD models. In: Proc. CVPR. pp. 3762–3769, 2014.

[Br16] Brachmann, Eric; Michel, Frank; Krull, Alexander; Ying Yang, Michael; Gumhold, Stefan; Rother, carsten: Uncertainty-Driven 6D Pose Estimation of Objects and Scenes From a Single RGB Image. In: Proc. CVPR. 2016.

[DT05] Dalal, Navneet; Triggs, Bill: Histograms of Oriented Gradients for Human Detection. In: Proc. CVPR. pp. 886–893, 2005.

[Gi16] Giusti, Alessandro et al.: A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots. IEEE Robotics and Automation Letters, 2016.

[He15] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian: Deep Residual Learning for Image Recognition. CoRR, abs/1512.03385, 2015.

[Ho16] Hoffman, Judy; Wang, Dequan; Yu, Fisher; Darrell, Trevor: FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. CoRR, abs/1612.02649, 2016.

[HOM15] Hueting, Moos; Ovsjanikov, Maks; Mitra, Niloy J.: CrossLink: Joint Understanding of Image and 3D Model Collections Through Shape and Camera Pose Variations. ACM TOG, 34(6):233:1–233:13, 2015.

[KSH12] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E: ImageNet Classification with Deep Convolutional Neural Networks. In: Adc. NIPS 25, pp. 1097–1105. 2012.

[Lo04] Lowe, David G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision, 60(2):91–110, November 2004.

[LSP06] Lazebnik, Svetlana; Schmid, Cordelia; Ponce, Jean: Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In: Proc. CVPR. pp. 2169–2178, 2006.

[LW15] Long, Mingsheng; Wang, Jianmin: Learning Transferable Features with Deep Adaptation Networks. CoRR, abs/1502.02791, 2015.

[Ru15] Russakovsky, Olga et al.: ImageNet Large Scale Visual Recognition Challenge. Proc. IJCV, 115(3):211–252, 2015.

[Sz15] Szegedy, Christian; Vanhoucke, Vincent; Ioffe, Sergey; Shlens, Jonathon; Wojna, Zbigniew: Rethinking the Inception Architecture for Computer Vision. CoRR, abs/1512.00567, 2015.

[Yo14] Yosinski, Jason; Clune, Jeff; Bengio, Yoshua; Lipson, Hod: How transferable are features in deep neural networks? CoRR, abs/1411.1792, 2014.

[Za16] Zamir, Amir Roshan; Wekel, Tilman; Agrawal, Pulkit; Wei, Colin; Malik, Jitendra; Savarese, Silvio: Generic 3D Representation via Pose Estimation and Matching. In: Proc. ECCV. pp. 535–553, 2016.

[ZF13] Zeiler, Matthew D.; Fergus, Rob: Visualizing and Understanding Convolutional Networks. CoRR, abs/1311.2901, 2013.

[Zh15] Zhang, Xucong; Sugano, Yusuke; Fritz, Mario; Bulling, Andreas: Appearance-based Gaze Estimation in the Wild. In: Proc. CVPR. pp. 4511–4520, 2015.