

---

# Estimating Chair Pose Using ConvNets

---

**Keunhong Park**  
University of Washington  
kpar@cs.washington.edu

**Aditya Sankar**  
University of Washington  
aditya@cs.washington.edu

## Abstract

The goal of our project is to align 3D models of chairs to 2D scene images containing chairs. In order to solve the alignment problem, we need to determine the translation, scale and rotation of the 3D object that maps it to the 2D image. For the purposes of this project we limit the scope to only azimuth orientation. We train a convolutional neural network on 64x64 rendered RGB chair images to predict the orientation of a previously unseen input.

Our convolutional neural network consists of two convolutional layers with max pooling and ReLU activation functions and two fully connected layers also with ReLU activation functions. This simple architecture yielded surprisingly good results giving us 99% precision on the testing set.

We evaluate the model on both a set of clean product images obtained from Bing and on cluttered natural images from the PASCAL VOC dataset and present a webcam interface that predicts chair orientation at interactive rates.

## 1 Introduction

We attempt to solve the 2D-3D alignment problem for objects in a given image. The general problem is described as follows: Given a single image and a large collection of 3D object models, can we find the optimal style and alignment (rotation, scale, translation) of a model that best explains the object seen in the image.

Since there are an arbitrarily large number of objects possible in a natural scene, we will initially focus our efforts on one object class: chairs. Chairs commonly occur in indoor scenes and exhibit interesting variance in style and function. Chairs are also convenient since there already exists a large number of 3D models online, which we can use to train our model.

For the purposes of this project, we model only the rotation around the vertical axis (azimuth) and train a classifier that predicts the azimuth given a previously unseen test example. This is fairly challenging by itself. Our approach should also generalize to elevation and roll angles, but we pick azimuth, since it is the most frequently varying component in chair images. Variation in elevation is generally limited since photographs are taken from eye level and roll is limited since the camera's vertical axis usually aligns with gravity. However, exceptions do exist, as we have observed in real data.

Our test data includes both withheld training data and natural images downloaded from the Internet. Within images downloaded from the Internet, we have sub-classes of images that have a clean background (product images from Bing.com) and images that are more naturally captured (PASCAL VOC)

We train a simple convolutional neural network to model the data. For training we utilize a publicly available dataset [1] of pre-rendered 3D chair models. Each rendered 600x600px image is labeled with  $\{chairtype, azimuth, elevation\}$ . Details of our data processing and model training are described in following sections.

We finally demonstrate that our trained model is able to detect the orientations of chairs from natural images and from a webcam stream at interactive rates.

## 2 Related Work

Our training data is obtained from Aubry et. al. [1], who use parts-based correspondence with discriminative histogram of gradients (HOG) features that are computed with linear discriminant analysis (LDA). Our approach differs in that we are training a convNet that learns the feature weights via backpropagation.

Huang et. al. [2] reconstruct a representation of the chair by jointly analyzing a product-catalog style image and a deformable 3D model. Their test images are restricted to well-cropped, noise free images whereas we intend to test our approach using convolution on natural images with more clutter and noise.

Satkin et. al. [3] and Hedau et. el. [4] use a comparable approach to model indoor scenes using a database of 3D models. While these systems work on more general scenes, they include several strong priors, such as visible vanishing points, manhattan alignment etc., in order to make assertions about the position and orientation of objects. Our model while restricted to a single object class, is designed to be more general.

Dosovitskiy et. al. [5] share insights on the inner workings of Neural Networks while analyzing chairs. By studying the output of hidden layers, they are able to generate new chair models that morph between training examples, but are otherwise previously unseen. The insight from this paper may inform the design of our neural network architecture.

## 3 Training Data

Our training data is obtained from [1] and consists of a set of pre-rendered chair models. There are 1398 distinct models with 31 discretized azimuth angles and 2 elevation angles, totaling 86,676 images. Of this, we withhold 18,476 images as our validation set.

The chairs are randomly shuffled and the images are cropped to remove any white-space. The images are then resized to 64x64 pixels, which was determined to be the optimal size for training that maintains image quality while converging in a reasonable time-frame (roughly 8 hours). The chairs are also whitened per-image so each image has zero mean and unit variance.

Finally, we additionally apply random brightness and contrast changes to each image. This is in order to account for varying image/lighting conditions and to artificially increase the data set size.



Figure 1: A sample of the 3D models we have available.

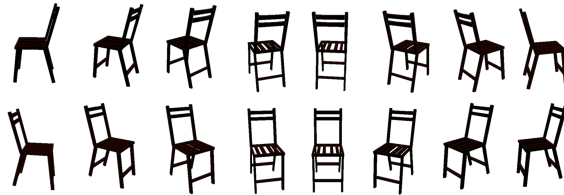


Figure 2: We render each model from 31 different azimuth angles and 2 elevation angles.

## 4 CNN Architecture

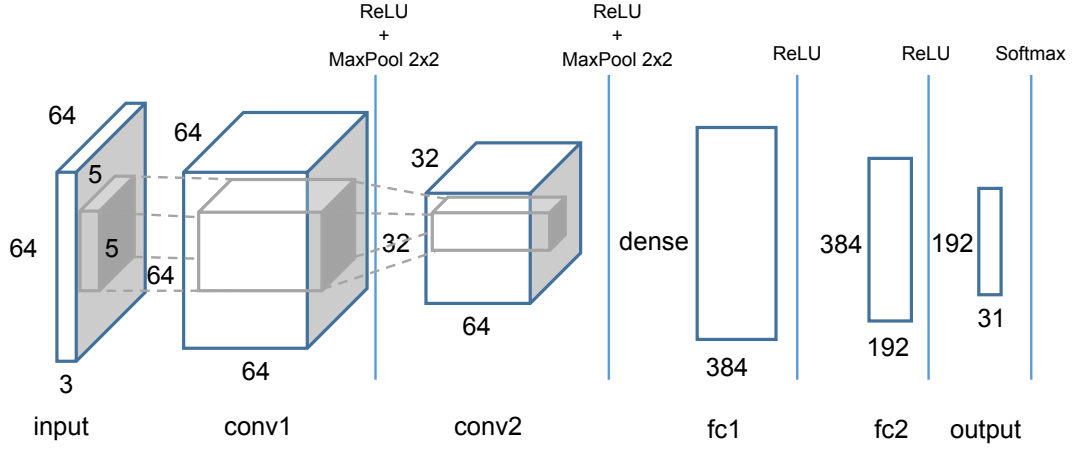


Figure 3: An illustration of our simple CNN architecture.

We use a simple convolutional neural network with a input layer of size 64x64x3 (our code allows for an input of any size which is then just shrunk by the max-pooling layer). The two convolutional layers have filters of size 5x5 and stride 1 with a depth of 64. Each convolutional layer is followed by a 3x3 max-pooling layer with stride 2 which shrinks the input size from in half (e.g. 64x64 to 32x32) and have rectified linear units (ReLU) as the non-linearity.

The ReLUs are simply given by:

$$f(x) = \max(0, x)$$

The above ReLU function has empirically been shown to work better than other non-linearities in some neural network tasks. [6]

The convolutional layers are followed by a dense connection to a fully connected layer. We have two fully connected layers which are also activated by ReLUs.

The final output layer outputs confidence scores for 31 different classes which each correspond to one azimuth angle.

We train the network end-to-end with the 64x64 rendered RGB image as an input and the azimuth class as an output with the loss being the cross entropy between the ground truth and the output. The loss of our network is thus given by the following:

$$H_y(\hat{y}) = \sum_i y_i \log \hat{y}_i$$

where  $y$  is the set of our ground truth labels and  $\hat{y}$  is the set of our predictions.

## 5 Model Training

We train our convolutional neural network on a set of 68,200 training images for 10000 iterations using a stochastic gradient descent optimizer with a batch size of 7936 with a learning rate of 0.01. We present our precision and loss by iteration in the following figures.

Our convolutional network was implemented with the recently released TensorFlow toolkit [7]. The hardware used for network training is an Intel Xeon(R) CPU X5680 @ 3.33GHz x 24 cores, 64-bit OS, 23.5 GiB of Memory, GeForce GTX 750 Ti/PCIe/SSE2 with 2048 MiB of graphics memory.

On the above hardware, our model takes about 8 hours to converge. Cross entropy and loss functions are plotted below.

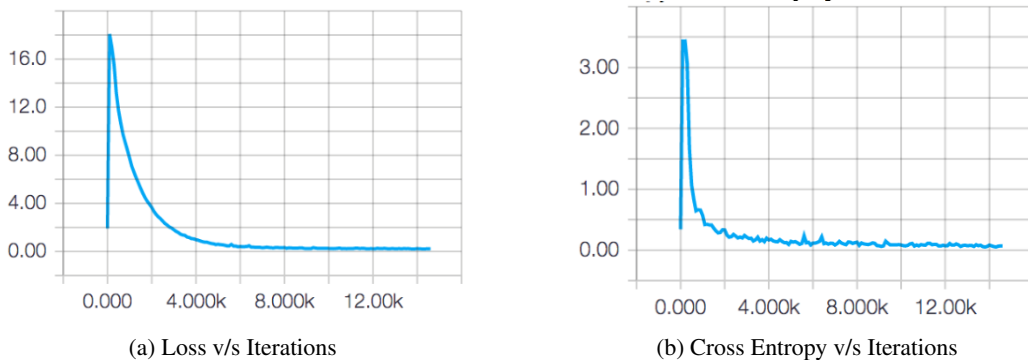


Figure 4: Plots for training output v/s Iterations

## 6 Results

### 6.1 Quantitative Results

We evaluate our method quantitatively on a withheld testing set of rendered synthetic models which have the same statistics as our training data and also on natural images of chairs gathered and cropped from the PASCAL VOC challenge data set. For the PASCAL VOC data, we crop all objects labeled ‘chair’ that are annotated as non-difficult, non-occluded and non-truncated. This gives us 424 chair images, which we then manually annotate with ground-truth azimuth angles.

|          | Precision |
|----------|-----------|
| Training | 0.99      |
| Testing  | 0.924     |

(a) Synthetic data

|                  | Precision |
|------------------|-----------|
| Top 1            | 0.1415    |
| Top 3            | 0.3184    |
| Incl. 1 adjacent | 0.2712    |

(b) PASCAL VOC Data

Table 1: Precision results on synthetic and natural image data.

While our precision statistics on PASCAL VOC data is significantly (order of magnitude) better than random chance, we still get relatively poor results on the dataset for several reasons.

**Different statistics from training set** Our training data is all cleanly cropped rendered chair images with white backgrounds. The PASCAL VOC dataset has very different statistics in that the background is cluttered, the images have noise, the chairs are not necessarily perfectly cropped, the lighting conditions may vary significantly, the chairs may be occluded by other objects including other chairs, and there is no constraint on the orientation of the chairs (i.e. they may contain extreme elevation and roll angles).



Figure 5: Examples of difficult images in the PASCAL VOC dataset.

## 6.2 Qualitative Results

We also evaluate our model qualitatively on a dataset of clean product images collected from Bing. These images are real images of chairs but have clean white backgrounds similar to our training set. We did not have ground truth annotations for these images thus could only do a qualitative examination but it looked to get over 90% of the orientations correct.



















































|                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |                                                                                   |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 0264.jpg                                                                          | Confidence/Azimuth                                                                | 4.6356 / 325                                                                      | 4.4407 / 336                                                                      | 3.6498 / 313                                                                      | 0132.jpg                                                                          | Confidence/Azimuth                                                                | 10.4791 / 301                                                                     | 9.8378 / 290                                                                      | 8.8787 / 313                                                                      |
|  |  |  |  |  |  |  |  |  |  |
| 0173.jpg                                                                          | Confidence/Azimuth                                                                | 7.1090 / 220                                                                      | 6.5536 / 209                                                                      | 4.3215 / 232                                                                      | 0225.jpg                                                                          | Confidence/Azimuth                                                                | 10.8890 / 278                                                                     | 9.2394 / 0                                                                        | 8.8550 / 11                                                                       |
|  |  |  |  |  |  |  |  |  |  |
| 0218.jpg                                                                          | Confidence/Azimuth                                                                | 8.7200 / 255                                                                      | 7.8649 / 267                                                                      | 6.1052 / 243                                                                      | 0288.jpg                                                                          | Confidence/Azimuth                                                                | 5.8699 / 232                                                                      | 5.7168 / 243                                                                      | 5.2002 / 255                                                                      |
|  |  |  |  |  |  |  |  |  |  |
| 0057.jpg                                                                          | Confidence/Azimuth                                                                | 12.4668 / 278                                                                     | 11.6173 / 290                                                                     | 9.5198 / 197                                                                      | 0259.jpg                                                                          | Confidence/Azimuth                                                                | 9.9799 / 232                                                                      | 9.6581 / 220                                                                      | 4.1981 / 243                                                                      |
|  |  |  |  |  |  |  |  |  |  |
| 0436.jpg                                                                          | Confidence/Azimuth                                                                | 9.8271 / 348                                                                      | 6.5376 / 0                                                                        | 6.2680 / 336                                                                      | 0301.jpg                                                                          | Confidence/Azimuth                                                                | 16.0943 / 232                                                                     | 11.2463 / 243                                                                     | 10.9415 / 220                                                                     |
|  |  |  |  |  |  |  |  |  |  |

Figure 6: Results of our method on clean product images obtained from Bing.

## 6.3 Detecting Chair Pose via Webcam

We have implemented a real-time system that detects the pose of a chair through a webcam video stream. The system crops and resizes the image and then evaluates it using the trained network model. The top 3 predicted orientation results are displayed pictorially with a generic chair model.

On the same hardware as we used for training, the webcam system runs at around 3 frames per second. Interactively rotating the chair reflects in the predicted results based on our model. The demo also has the capability to run over RPC, so that a thin client can provide the webcam input and the evaluation can be performed on a server, without sacrificing performance. Screenshots of the interactive application are shown below.

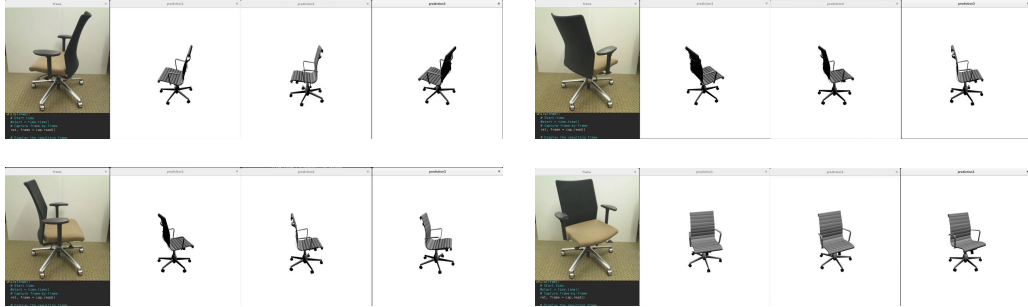


Figure 7: A chair captured in four different orientations via a webcam. On the right, our system outputs the top three predictions for the azimuth orientation of the input chair.

## 7 Discussion and Future Work

Obtaining ground truth 3D labeling for natural images is hard. However, using synthetic 3D models rendered from various angles can provide sufficient training data for this problem.

With a relatively simple two layer neural network, our model is able to predict the correct azimuth class with 92.4% (17072 out of 18476 images) on the withheld data. It also performs significantly better than guessing on PASCAL VOC data, although the results are not as accurate as clean product images.

There are several improvements that should enhance the performance on real image data. The most important improvement would be to train classifiers for elevation and roll angles as well. By adding random backgrounds (of natural scenes, like homes, offices, outdoors) we can attempt to remove the bias towards clean backgrounds. By introducing multi-scale training images, we can capture the scale of the object (translation is already accounted for by the convolution).

Finally, training a classifier to detect the chair type (appearance) would be very useful. However, this is a much more challenging problem since in our training set we have 1398 distinct chairs. Correctly predicting 1 out of 1398 can be prone to noise. However, one possible solution is to create sub-classes of chair types (Office chairs, sofas, dining room chairs etc.) and detect type within each sub-class.

## References

- [1] Mathieu Aubry, Daniel Maturana, Alexei A. Efros, Bryan C. Russell, and Josef Sivic. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14*, pages 3762–3769, Washington, DC, USA, 2014. IEEE Computer Society.
- [2] Qixing Huang, Hai Wang, and Vladlen Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Trans. Graph.*, 34(4):87:1–87:10, July 2015.
- [3] Scott Satkin, Jason Lin, and Martial Hebert. Data-driven scene understanding from 3D models. In *Proceedings of the 23rd British Machine Vision Conference*, 2012.
- [4] Varsha Hedau, Derek Hoiem, and David Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *Proceedings of the 11th European Conference on Computer Vision: Part VI, ECCV'10*, pages 224–237, Berlin, Heidelberg, 2010. Springer-Verlag.
- [5] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. *CoRR*, abs/1411.5928, 2014.
- [6] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, 2013.
- [7] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.