

Learning to Generate Chairs, Tables and Cars with Convolutional Networks

By Alexey Dosovitskiy, Jost Tobias Springenberg,
Maxim Tatarchenko, Thomas Brox

Presentation by Sam Woolf and Cole Springate

Overview

Generative models often have two components:

1. Learn to represent the distribution of images
2. Learn to generate an image from a vector drawn from that distribution

Paper focuses on 2. High level descriptions are given to train generator.

Task

Generate images from their high-level descriptions

Dataset: projections of 3D images and their high level descriptions

Network: descriptions  2D projection

High level description includes

1. Style
2. Viewpoint
3. Numerous transformation parameters (color, zoom)

Task: more than memorization

Network should learn a presentation of 3D models and be able to:

- Transfer knowledge within a class
- Transfer knowledge between classes
- Interpolate within and between classes
- Create new images not seen by network

Related work

Other generative strategies:

GANS (generative adversarial networks)

- We saw this in “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks” (presented by Jorge)

Restricted Boltzman Machines

- We saw this in “Using very deep autoencoders for content-based image retrieval” (presented by Chris and Ben)

Paper contribution (differences)

1. Assume high level latent representation is given
2. Use supervised training

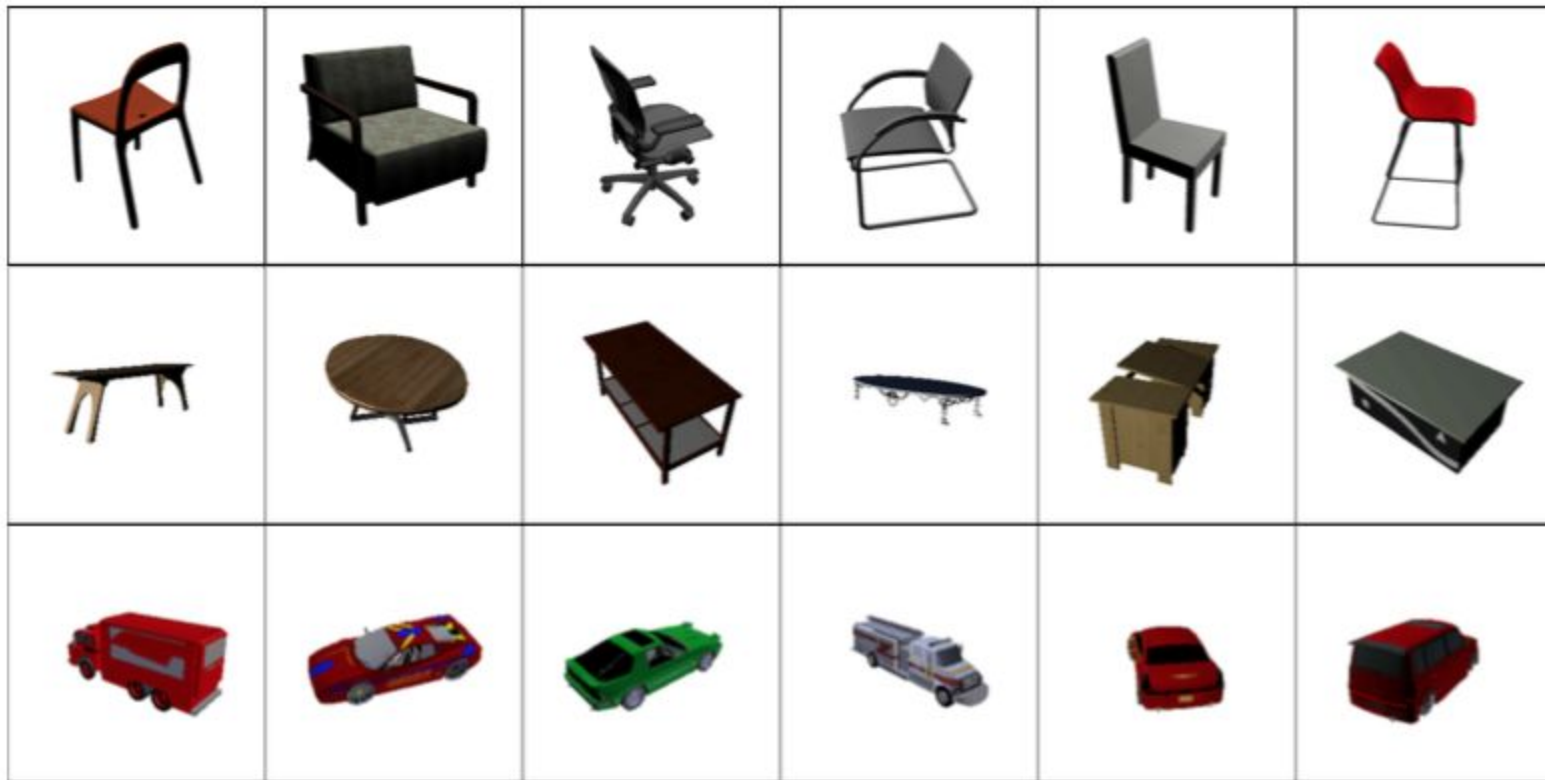
Upside:

- relatively high quality images (up to 256 x 256)
- control over which images to generate

Downside:

- need to start with high level representation

Dataset



Dataset

Chair Model Renderings

- 809 models
- each with 2 elevation angles and 31 azimuth angles
- segmentation mask produced by subtracting white background

Cars and Table Models

- 7124 car models, 1000 table models
- rendered by authors (models from ShapeNet)
- 5 elevation angles and 36 azimuth angles
- segmentation rendered directly

Dataset size and augmentation

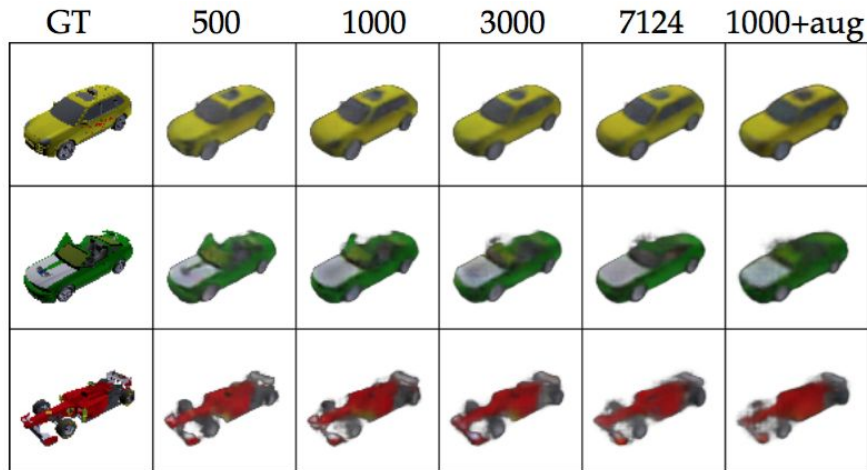


Fig. 5. Qualitative results for different numbers of car models in the training set.

Num models	500	1000	3000	7124	1000aug
MSE ($\cdot 10^{-3}$)	0.48	0.66	0.84	0.97	1.18

TABLE 2

Per-pixel mean squared error of image generation with varying number of car models in the training set.

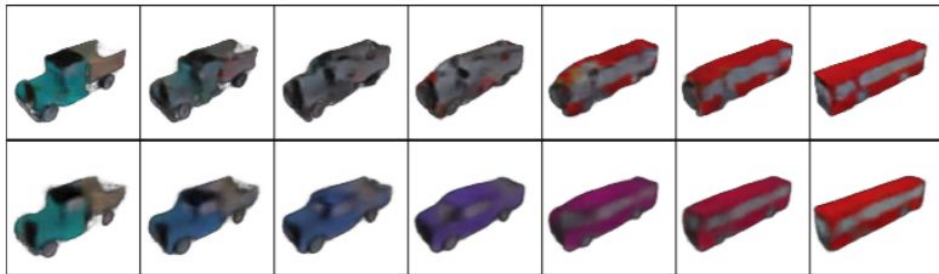


Fig. 6. Interpolation between two car models. **Top:** without data augmentation, **bottom:** with data augmentation.

Model - Input

$$D = \{(\mathbf{c}^1, \mathbf{v}^1, \boldsymbol{\theta}^1), \dots, (\mathbf{c}^N, \mathbf{v}^N, \boldsymbol{\theta}^N)\}$$

c: ones-hot encoding of the model (style)

v: azimuth and elevation of the camera (as sine and cosine)

θ : parameters of artificial transformations (color, brightness, zoom, etc)

Model - Output

$$\mathcal{O} = \{(\mathbf{x}^1, \mathbf{s}^1), \dots, (\mathbf{x}^N, \mathbf{s}^N)\}$$

MODEL

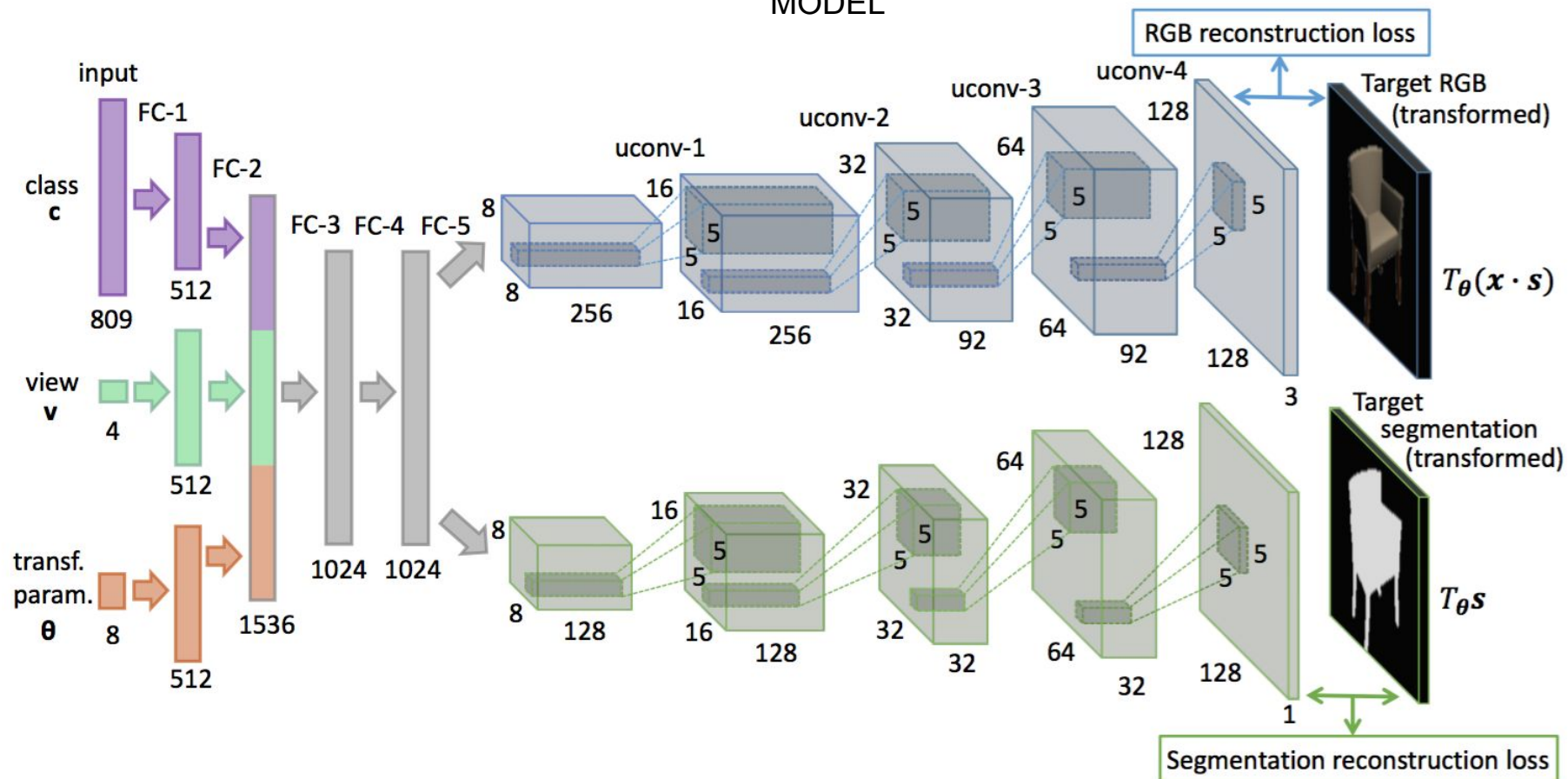
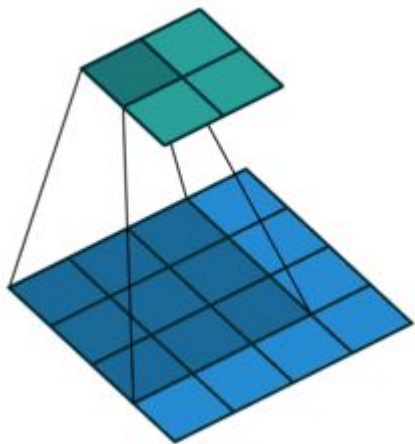


Fig. 1. Architecture of a 2-stream network that generates 128×128 pixel images. Layer names are shown above: FC - fully connected, uconv - unpooling+convolution.

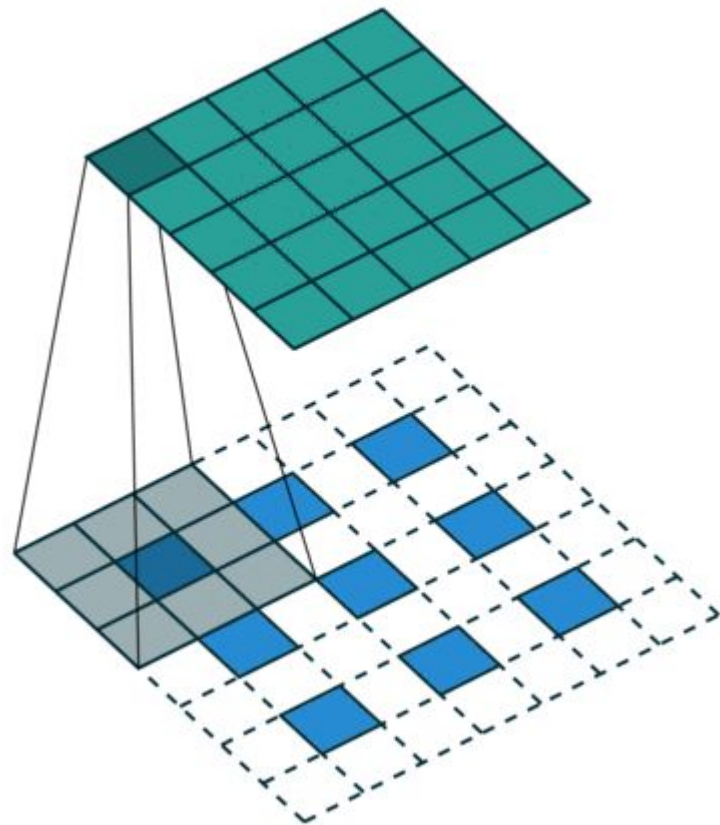
What is “Up-convolution”

Like “**deconvolution**” (actually called transposed convolution with fractional slide).

Regular convolution:



“Deconvolution”:



What is “Up-convolution”

Up-convolution is “unpooling” (opposite of max pooling) and then convolution:

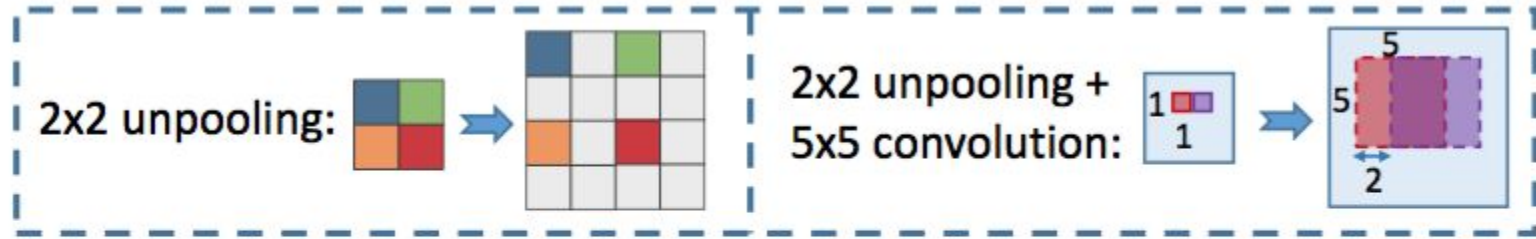


Fig. 2. Illustration of unpooling (left) and unpooling+convolution (right) as used in the generative network.

Training

Loss

- L2 for image
- softmax layer and negative log-likelihood for segmentation

Update

$$\min_{\mathbf{W}} \sum_{i=1}^N L_{RGB} (T_{\theta^i}(\mathbf{x}^i \cdot \mathbf{s}^i), u_{RGB}(h(\mathbf{c}^i, \mathbf{v}^i, \theta^i))) \\ + \lambda \cdot L_{segm} (T_{\theta^i} \mathbf{s}^i, u_{segm}(h(\mathbf{c}^i, \mathbf{v}^i, \theta^i))),$$

MODEL ALTERNATIVES

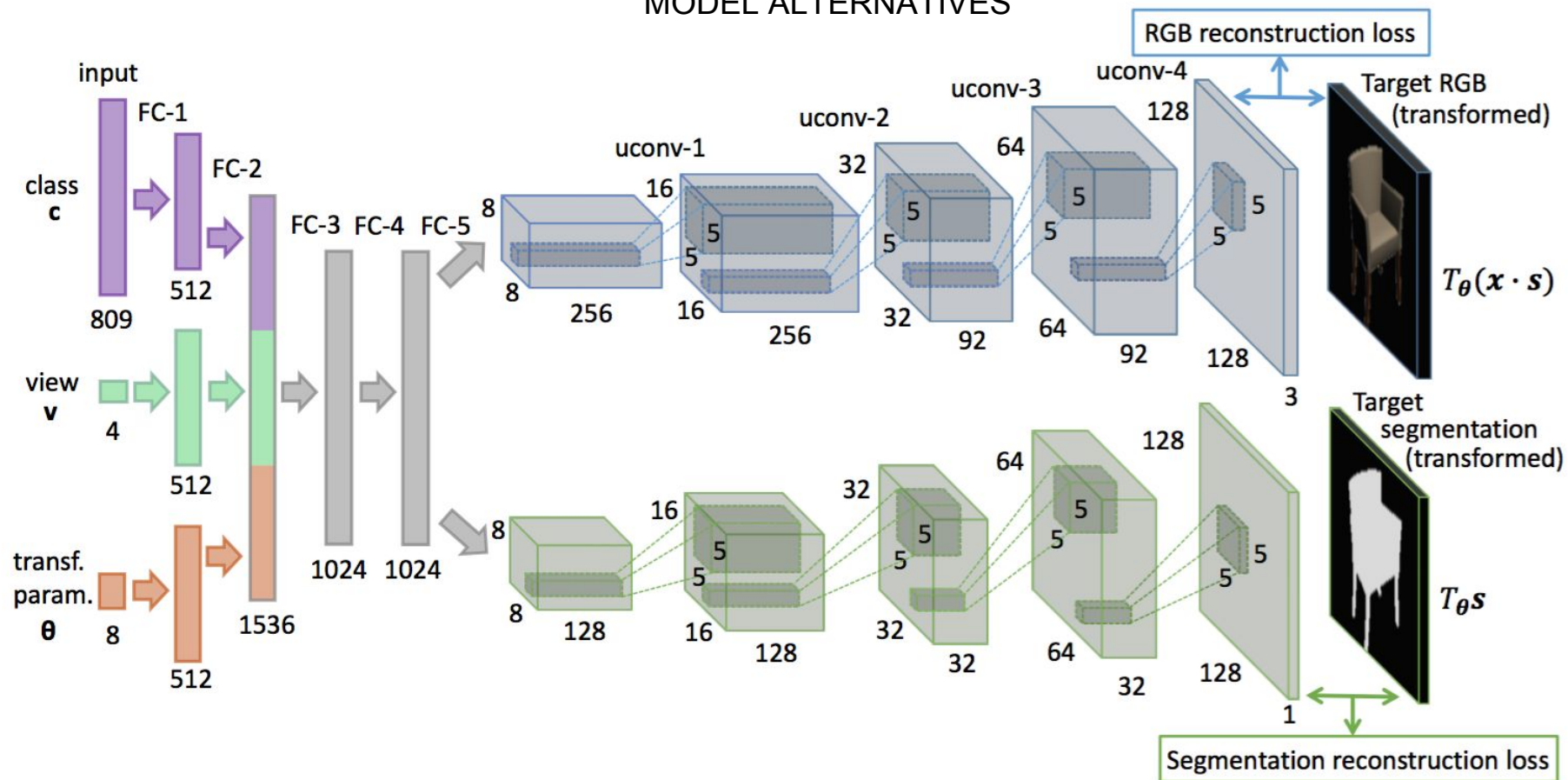


Fig. 1. Architecture of a 2-stream network that generates 128×128 pixel images. Layer names are shown above: FC - fully connected, uconv - unpooling+convolution.

Alternative Model Results:

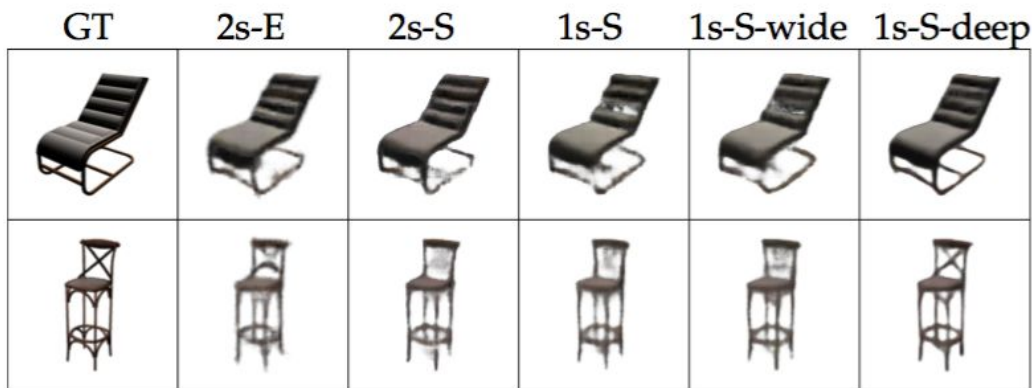


Fig. 4. Qualitative results with different networks trained on chairs. See the description of architectures in section 4.2.

Net	2s-E	2s-S	1s-S	1st-S-wide	1st-s-deep
MSE ($\cdot 10^{-3}$)	3.43	3.44	3.51	3.41	2.90
#param	27M	27M	18M	23M	19M

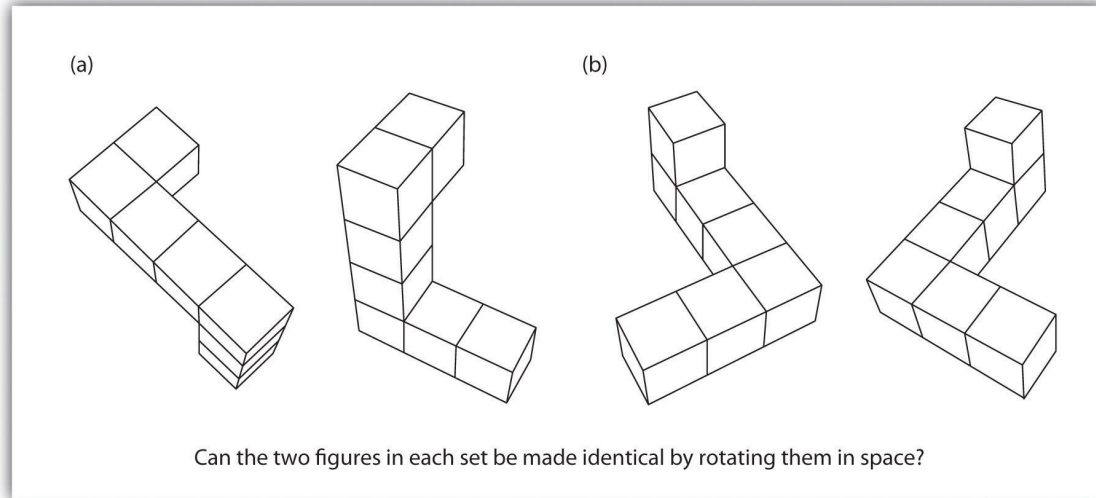
TABLE 1

Per-pixel mean squared error of the generated images with different network architectures and the number of parameters in the expanding parts of these networks.



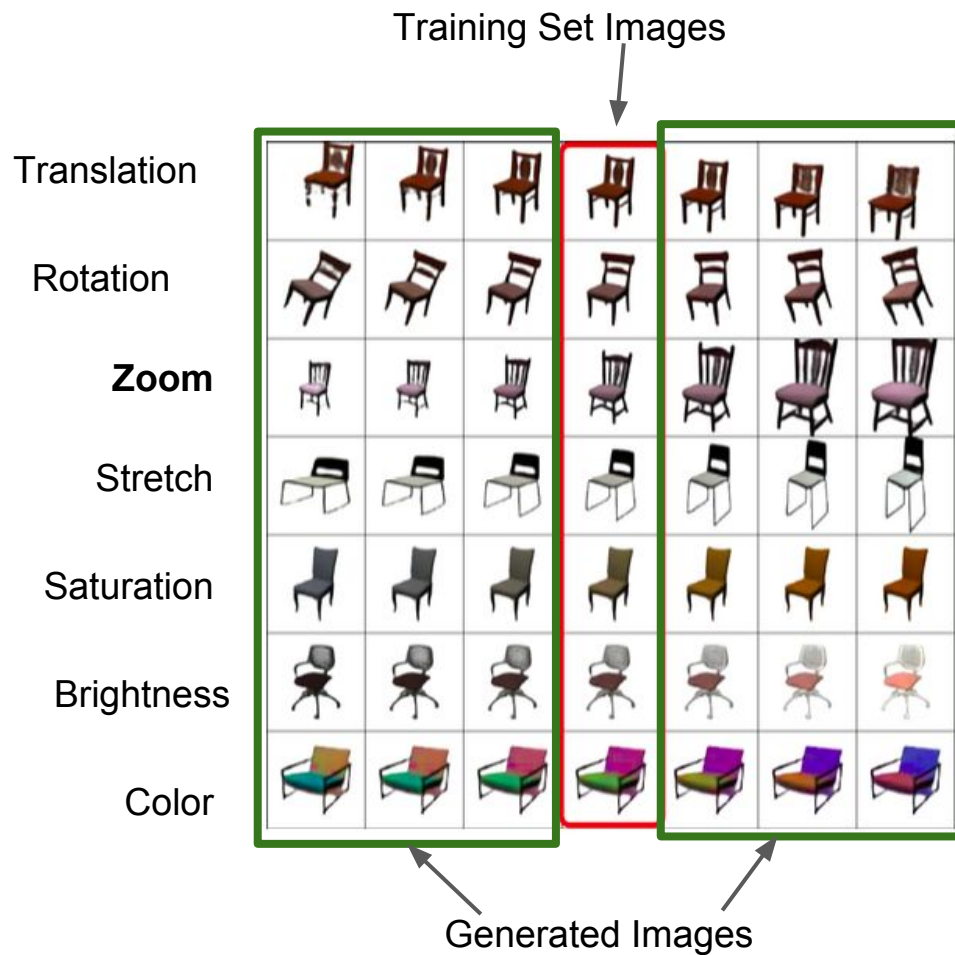
Best

Key Experimental Results



https://saylordotorg.github.io/text_introduction-to-psychology/section_13/7b058245d52f3821cd0727530bc761a1.jpg

Generating Transformations



View Interpolation

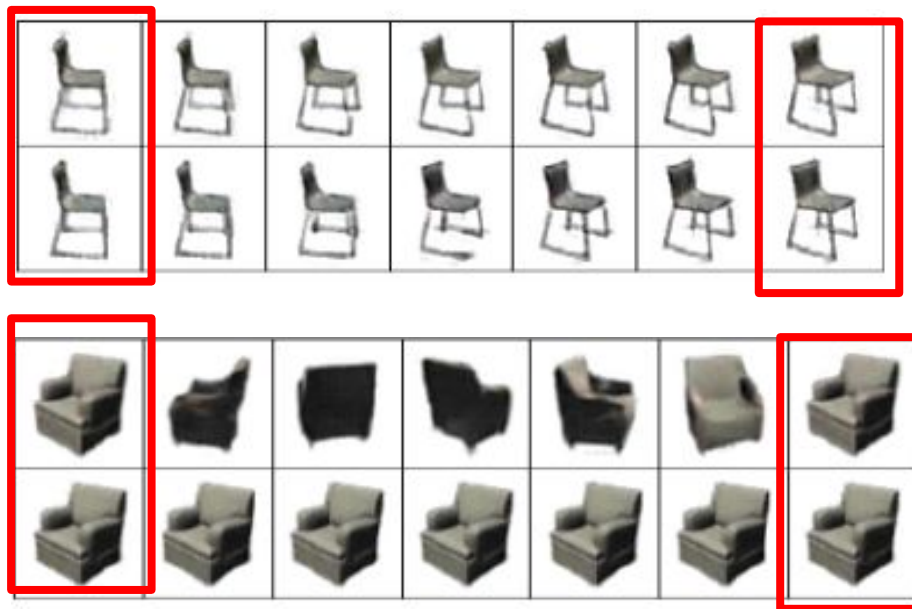
 = Training Set

Top row: w/ Knowledge Transfer
Bottom row: w/o Knowledge Transfer

Views of Chair
in training set

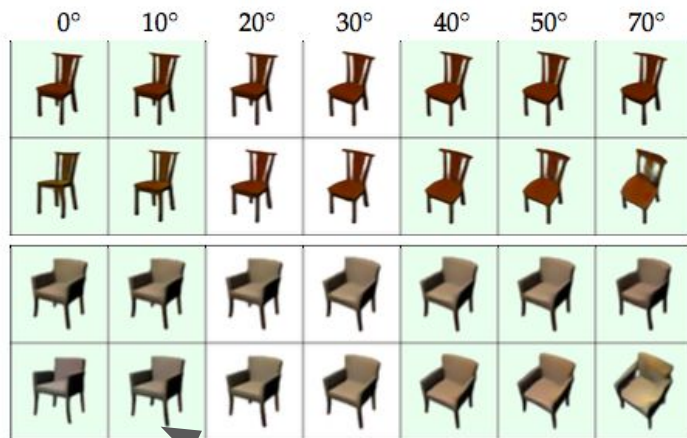
15

1

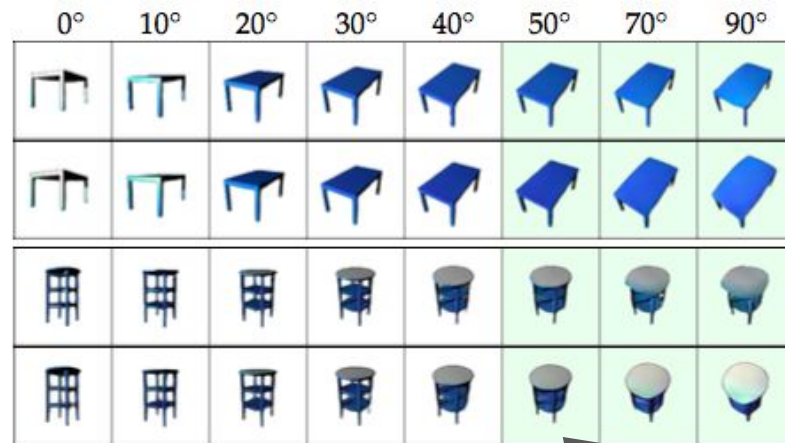


Elevation Angle Knowledge Transfer

1st Row: Trained on Chairs
2nd Row: Trained on Tables and Chairs



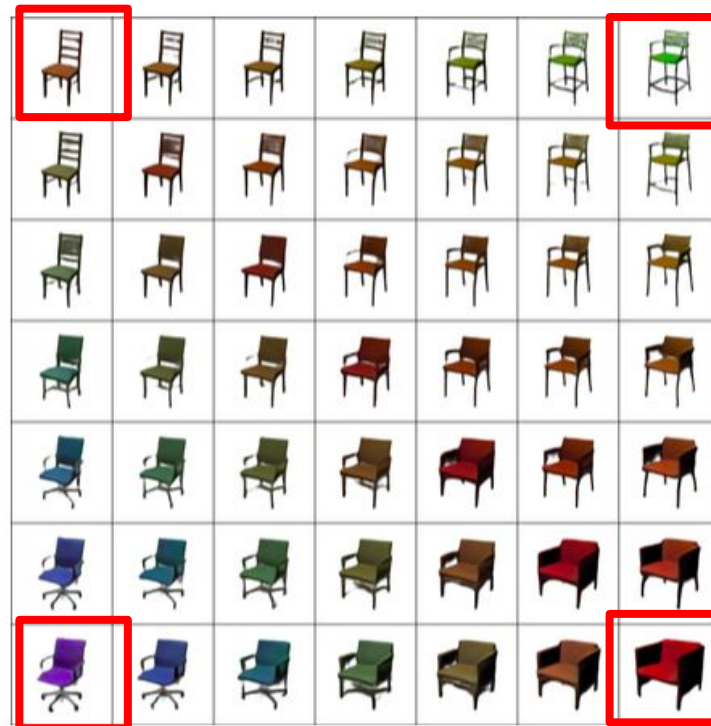
1st Row: Trained on Tables
2nd Row: Trained on Tables and Chairs



Generated

Generation w/ Style Interpolation

 = Training Set



Generation w/ Style Interpolation Cont.



 = Training Set

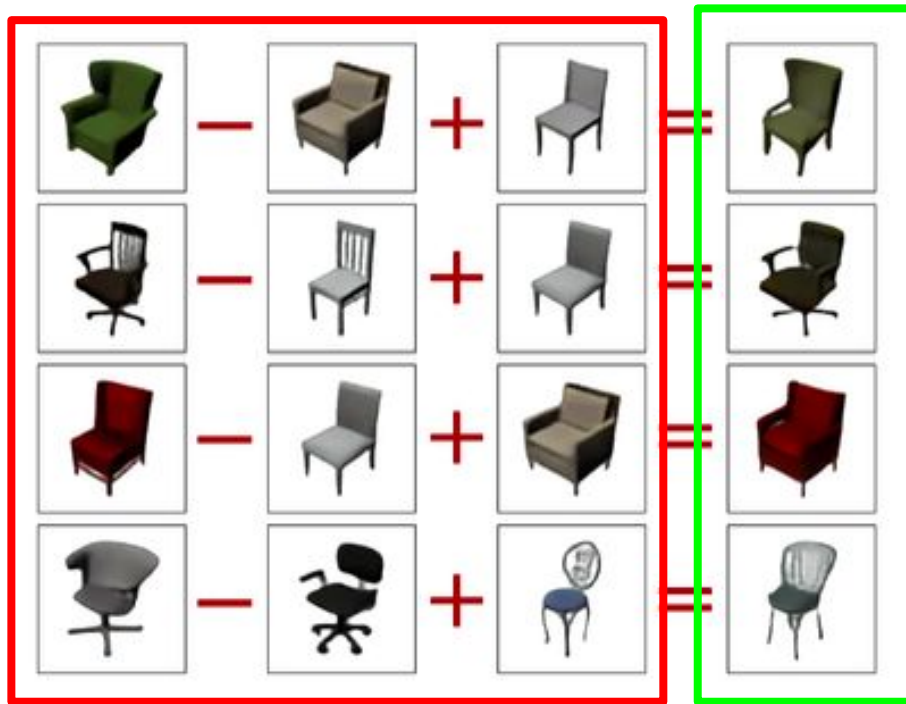
Chairs



Tables

Feature Arithmetic on FC-2

 = Training Set
 = Generated



Randomly Generated Chairs

Top Row: Generated Image
Bottom Row: Nearest Match in Training set

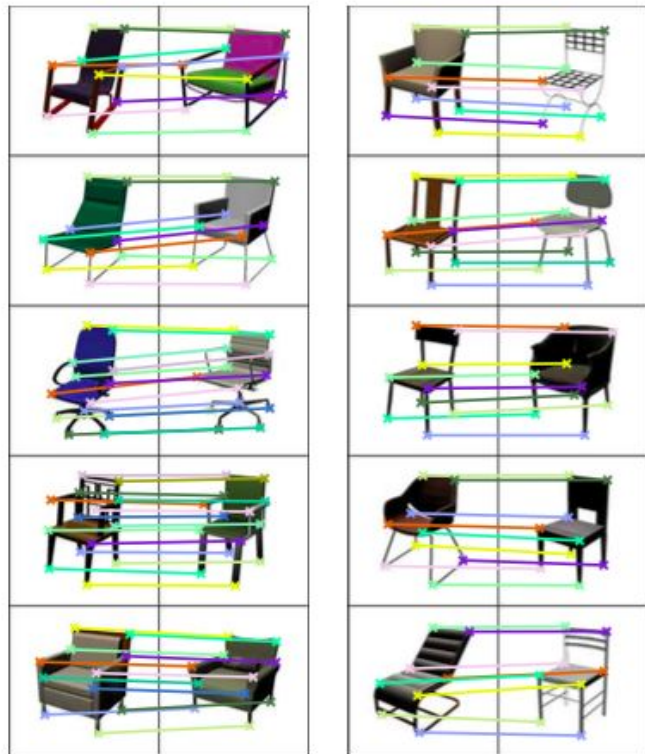


From Softmax of random (gaussian) input



From Gaussian noise in FC-2 of networks
trained with usual loss

Correspondences



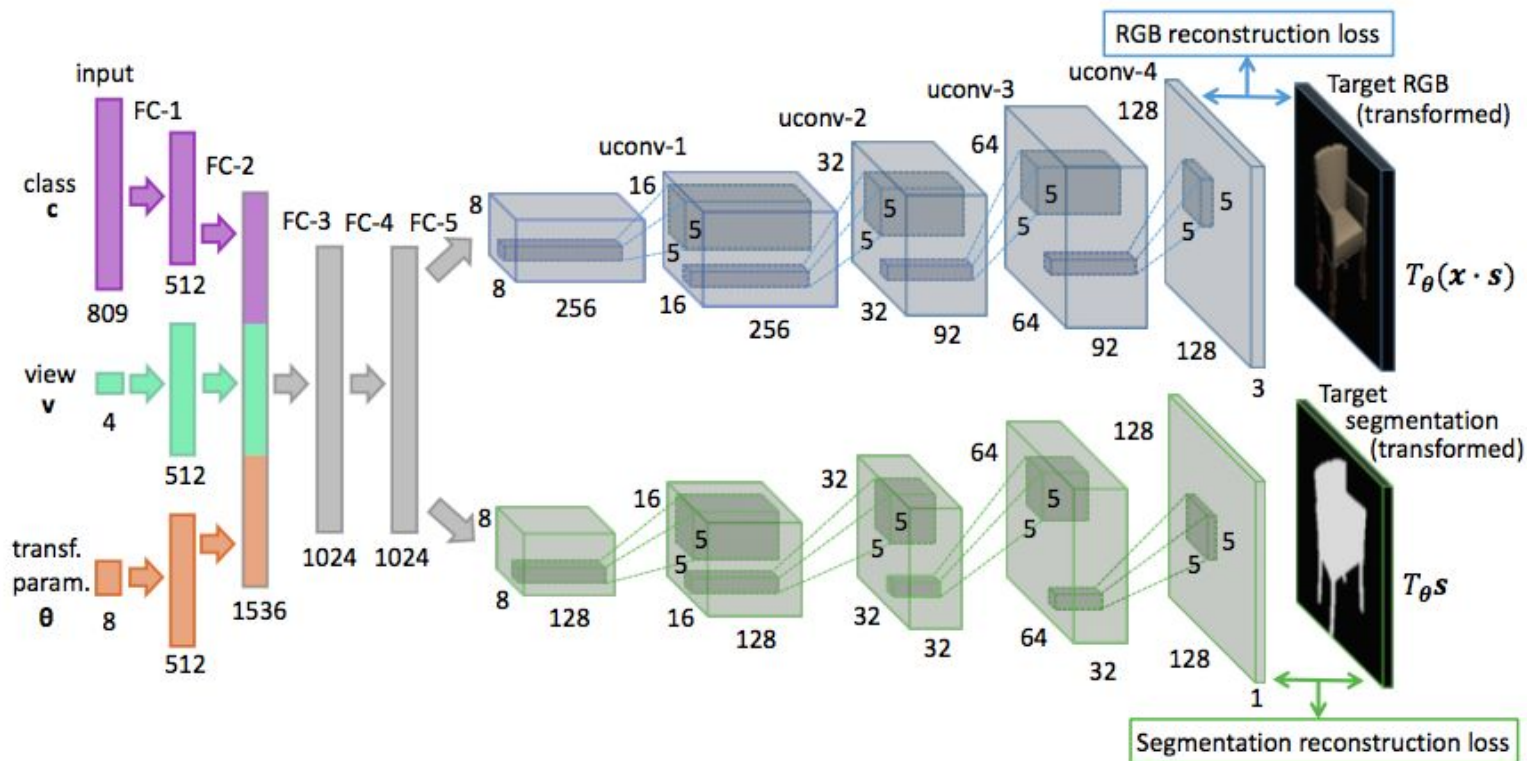
- Given 2 images in dataset, generate 64 images via interpolation
- Use optical flow to match keypoints

Method	All	Simple	Difficult
DSP [36]	5.2	3.3	6.3
SIFT flow [35]	4.0	2.8	4.8
Ours	3.4	3.1	3.5
Human	1.1	1.1	1.1

TABLE 3
Average displacement (in pixels) of corresponding keypoints found by different methods on the whole test set and on the 'simple' and 'difficult' subsets.

Analysis of the Network

A little reminder...

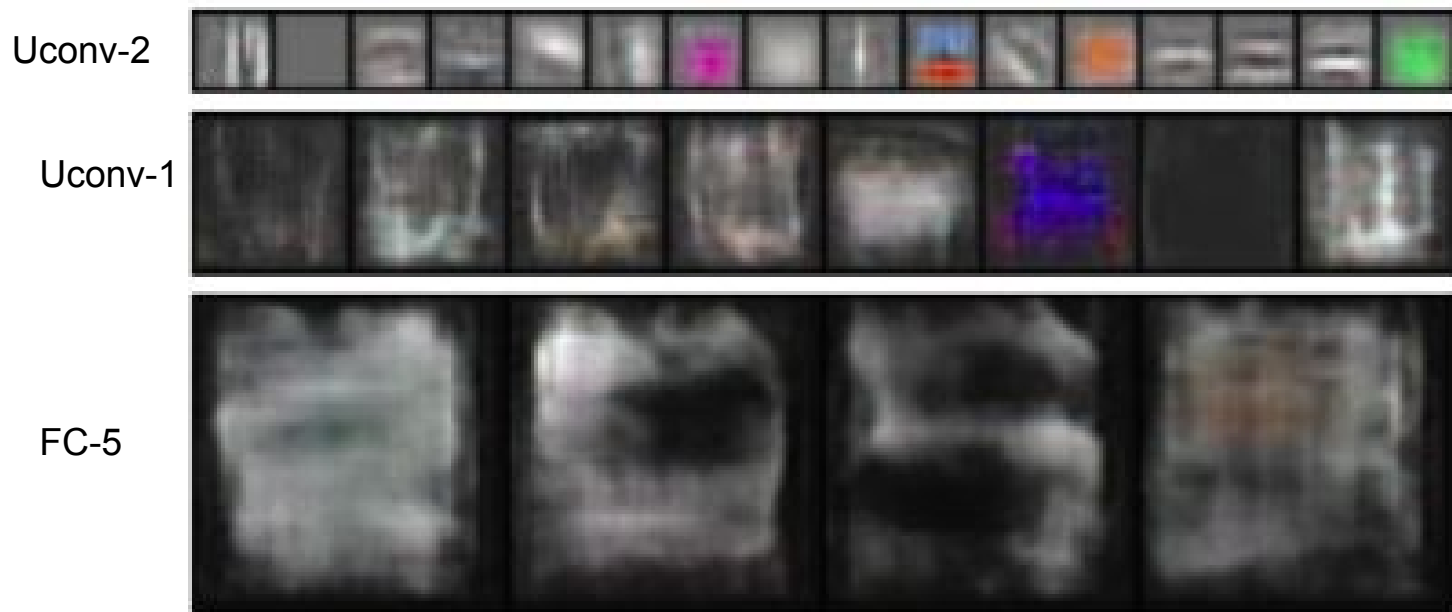


Images Generated from FC-4 Activations

translation upwards
Zoom
stretch horizontally
stretch vertically
rotate counter-clockwise
rotate clockwise
increase saturation
decrease saturation
make violet.



Images Generated from Single Neuron in Feature Map



Images Generated from Neighboring Neurons

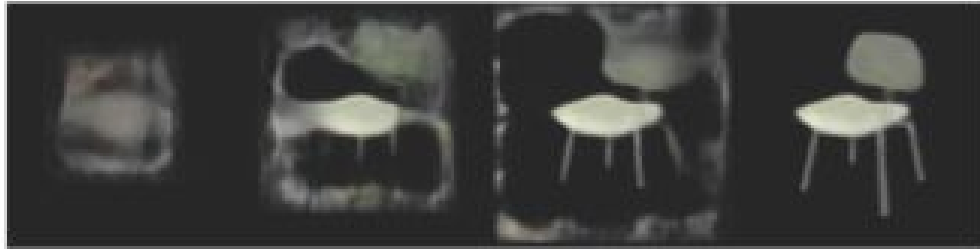


Fig. 24. Chairs generated from spatially masked FC-5 feature maps (the feature map size is 8×8). The size of the non-zero region increases left to right: 2×2 , 4×4 , 6×6 , 8×8 .

Conclusion

- A supervised CNN is used to generate images based on high-level information
- The network learns more than generate 2D samples, it learns a 3D representation
- One can use the network to invent new designs, based on a randomized input vector

Sources/Citations

- <https://web.stanford.edu/class/cs331b/presentations/paper6.pdf>
- https://en.wikipedia.org/wiki/Restricted_Boltzmann_machine
- https://github.com/stokasto/caffe/blob/chairs_deconv/README_chairs
- <https://arxiv.org/pdf/1411.5928.pdf>
- <https://www.youtube.com/watch?v=QCSW4isBDL0>
- <https://lmb.informatik.uni-freiburg.de/Publications/2015/DB15/>