Deep Convolutional Inverse Graphics Network

Presenter: Zhe Wang

https://qdata.github.io/deep2Read

Tejas D.Kulkarni, Will Whitney, Pushmeet Kohli, Joshua B.Tenenbaum

201909

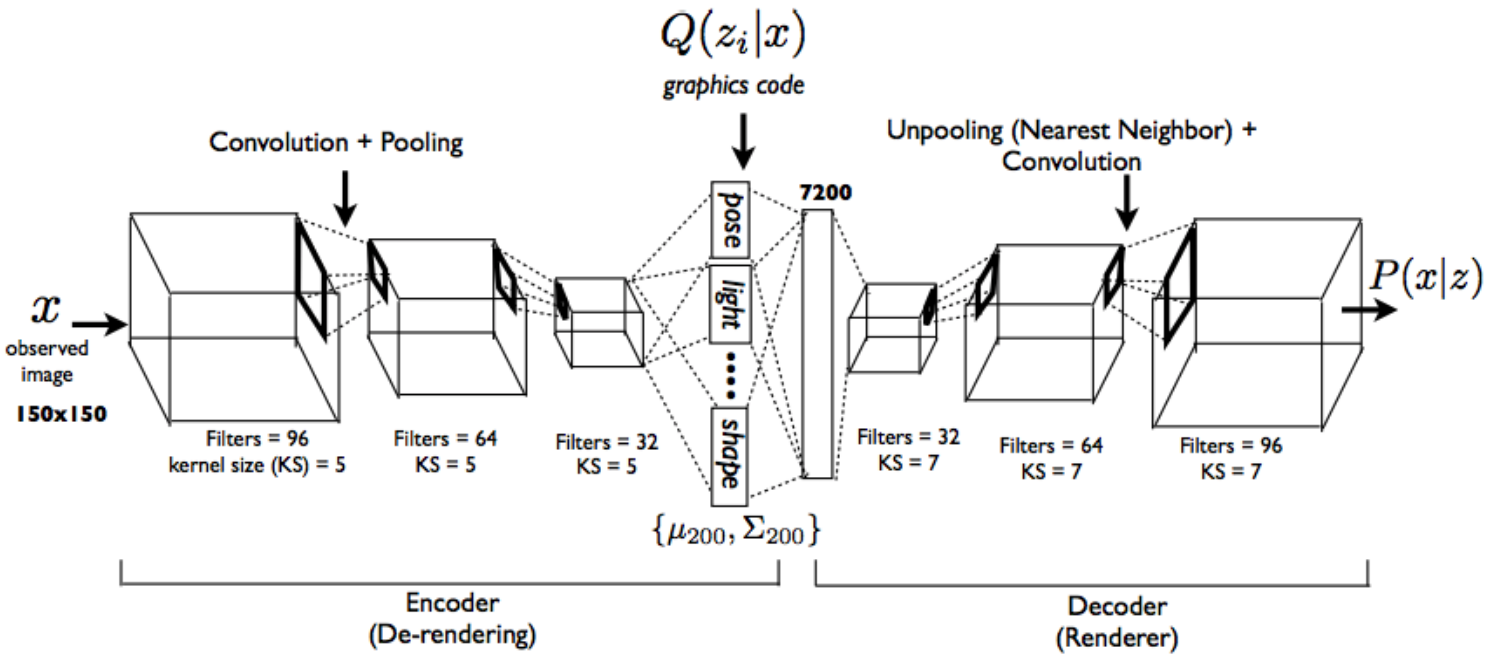
Target: Learning an interpretable representation of images in unsupervised manner.

Model architecture: VAE-based model

Solution: Proposing a training procedure to encourage each group of neurons in the code layer to distinctly represent a specific transformation

Vision as inverse graphics: Computer graphics consists of a function to go from the graphics code to images, and this graphics code is typically disentangled to allow for rendering scenes with fine-grained control over transformations such as object location, pose, lighting, texture, and shape.

Disentanglement representation learning is the inverse of computer graphics.



- Encoder network captures distribution over graphics codes Z given data x. $q(z|x)$
- Decoder network learns a conditional distribution to produce an approximation $\hat{x}$ given z. $p(x|z)$

Goal of this work is to learn a representation of the data which consists of disentangled and semantically interpretable latent variables. We would like only a small subset of the latent variables to explain these semantical changes, the remaining variables represent intrinsic properties.

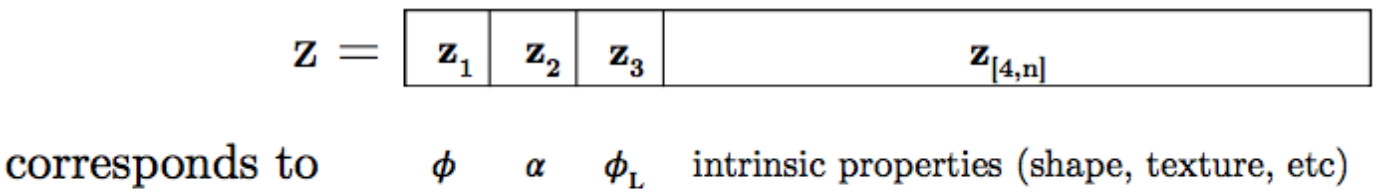


Figure 2: **Structure of the representation vector.** ϕ is the azimuth of the face, α is the elevation of the face with respect to the camera, and ϕ_L is the azimuth of the light source.

Training procedure:

Data collection:

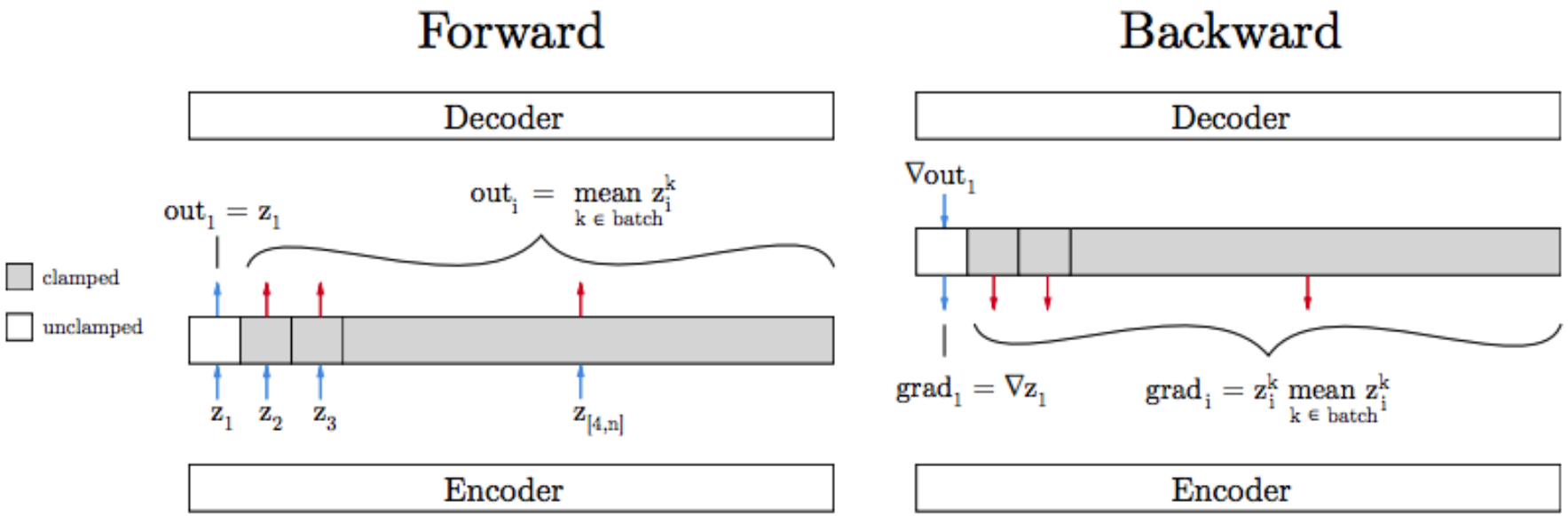
- Organize our data into mini-batches corresponding to changes in only a single scene variable (azimuth angle, elevation angle, azimuth angle of the light source)
- Generate mini-batches in which the three extrinsic scene variables are held fixed but all other properties of the face change.
- These mini-batches varying intrinsic properties are interspersed stochastically with those varying the extrinsic properties.

Training Procedure:
Forward:

- Select at random a latent variable z_{train} which we wish to correspond to one of azimuth angle, elevation angle, azimuth of light source, intrinsic properties.
- Select at random a mini-batch in which that only that variable changes.
- Show the network each example in the minibatch and capture its latent representation for that example z
- Calculate the average of those representation vectors over the entire batch.
- Before putting the encoder's output into the decoder, replace the values $z_i \neq z_{train}$ with their averages over the entire batch. These outputs are "clamped".

Backward:

- Calculate reconstruction error and backpropagate in the decoder.
- Use error gradients for those $z_i \neq z_{train}$. The gradient at z_{train} is passed through unchanged.
- Continue backpropagation through the encoder using the modified gradient.



- By clamping the output of all but one of the neurons, the decoder is forced to recreate all the variation in that batch using only the changes in that one neuron's value.
- By clamping the gradients, the encoder is forced to put all the information about the variations in the batch into one output neuron.

Experiments

Data generation: 12,000 batches of faces generated from a 3D face model, where each batch consists of 20 faces with random variations on face identity variables (shape/texture), pose, or lighting.

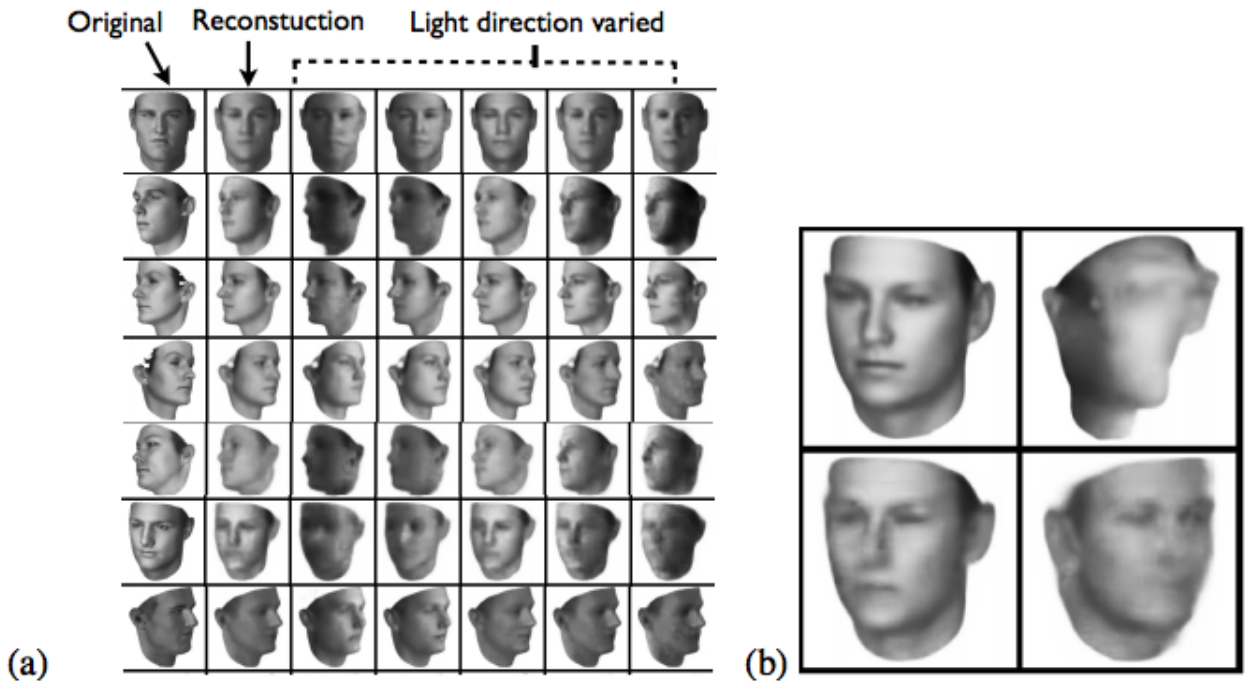


Figure 5: (a) **Manipulating light variables:** Qualitative results showing the generalization capability of the learnt DC-IGN decoder to render original static image with different light directions. The latent neuron z_{light} is changed to random values but all other latents are clamped. (b) **Entangled versus disentangled representations. Top:** Original reconstruction (left) and transformed (right) using a normally-trained network. **Bottom:** The same transformation using the DC-IGN.