Deep Convolutional Inverse Graphics Network
How did they do this?  Why did they do this?

How well did they do?  Can we do better?
How did they do this?

1. What model did they use?
2. What data did they use?
3. What training method did they use?
The Model – Variational AutoEncoder (VAE)
The Model – Variational AutoEncoder (VAE)
The Data – Synthetic, Feature Isolating Mini Batches

- All the images in a mini-batch only vary by a single parameter.
- Using synthetic data allows the authors to guarantee it only varies by a single parameter.
- Generate a large training set.
The Training Method – Clamping

1. Select a feature you want to isolate like direction of lighting or rotation of face. Call this feature $z_{\text{train}}$.

2. Select at random a mini-batch in which that only that variable changes.

3. Show the network each example in the minibatch and capture its latent representation for that example $z^k$.

4. Calculate the average of those representation vectors over the entire batch.
5. 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”.

6. Calculate reconstruction error and backpropagate as per SGVB in the decoder.

7. Replace the gradients for the latents \( z_i \neq z_{train} \) (the clamped neurons) with their difference from the mean (see Section 3.2). The gradient at \( z_{train} \) is passed through unchanged.

8. Continue backpropagation through the encoder using the modified gradient.
Do they average across features or across images in the batch?

When training on a mini-batch that isolates the lighting direction, what is the ideal variance of face elevation feature?

For the clamped features, why do they replace the gradients with the features differences from the mean?

After we swap the gradient with the difference from the mean do we use the same learning rate?
Do they average across features or across images in the batch?

Across images in the batch.

When training on a mini-batch that isolates the lighting direction, what is the ideal variance of face elevation feature?

For the clamped features, why do they replace the gradients with the features differences from the mean?

After we swap the gradient with the difference from the mean do we use the same learning rate?
Do they average across features or across images in the batch?

Across images in the batch.

When training on a mini-batch that isolates the lighting direction, what is the ideal variance of face elevation feature?

0 variance. The face elevation is not changing at all across the batch.

For the clamped features, why do they replace the gradients with the features differences from the mean?

After we swap the gradient with the difference from the mean do we use the same learning rate?
Do they average across features or across images in the batch?

Across images in the batch.

When training on a mini-batch that isolates the lighting direction, what is the ideal variance of face elevation feature?

0 variance. The face elevation is not changing at all across the batch.

For the clamped features, why do they replace the gradients with the features differences from the mean? To discourage the feature from varying across a batch that has no face elevation variance.

After we swap the gradient with the difference from the mean do we use the same learning rate?
Do they average across features or across images in the batch?

Across images in the batch.

When training on a mini-batch that isolates the lighting direction, what is the ideal variance of face elevation feature?

0 variance. The face elevation is not changing at all across the batch.

For the clamped features, why do they replace the gradients with the features differences from the mean? To discourage the feature from varying across a batch that has no face elevation variance.

After we swap the gradient with the difference from the mean do we use the same learning rate? No they use a much smaller learning rate. 1/100 of the original learning rate.
Why disentangle feature representations?

1. Interpretability
2. Robustness (Reusable)
3. Data Augmentation
Image Feature Arithmetic

\[
\text{smiling woman} - \text{neutral woman} + \text{neutral man} = \text{smiling man}
\]
Image Feature Arithmetic
Visual Question Answering Demo

http://vqa.cloudcv.org/
How well did they do?

1. Strengths
   a. Targets very specific features
   b. They can measure how much it isolated each feature
   c. Can learn to manipulate input images

2. Limitations
   a. How well would this method hold up on non-synthetic data?
   b. Generated images are fairly blurry. None of the images are looking directly forward.
Original face was in the training set
Original chair (leftmost column) was not in training set
Original chair (leftmost column) was not in training set
Can we do better?

1. Use a larger more network that can better represent the domain of images
2. Find a way to learn disentangled features without having to manually sort images into mini-batches.
1. Learning to generate chairs. Showed that you could make a generative network using continuous latent variables as input.
Past Related Works

1. Learning to generate chairs. Showed that you could make a generative network using continuous latent variables as input.
Future Related Works

1. InfoGAN (2016)
   a. Use Generative Adversarial Network with an imposed prior to encourage the generative network to learn disentangled feature representations
   b. Unsupervised
   c. Doesn’t take input images.
InfoGAN on MNIST

(a) Varying $c_1$ on InfoGAN (Digit type)
InfoGAN on MNIST

(c) Varying $c_2$ from $-2$ to $2$ on InfoGAN (Rotation)
InfoGAN on MNIST

(d) Varying $c_3$ from $-2$ to $2$ on InfoGAN (Width)
InfoGan on Chair Dataset

Notice that width is entangled with type of chair
Reference Links

Website for DC-IGN paper: http://willwhitney.github.io/dc-ign/www/


Visual Question Answering Demo: http://vqa.cloudcv.org/

InfoGAN: https://wiseodd.github.io/techblog/2017/01/29/infogan/

Facial Arithmetic: https://github.com/Newmu/dcgan_code
Summary

- Clamping and feature isolating mini-batches can distentangle the feature representations.
- DC-IGN can be used to manipulate images in a controlled way.
- Sorting non-synthetic training images into mini-batches may prove difficult.
- Unsupervised methods for disentangling feature representations exist. You can impose learning independent features as a prior.