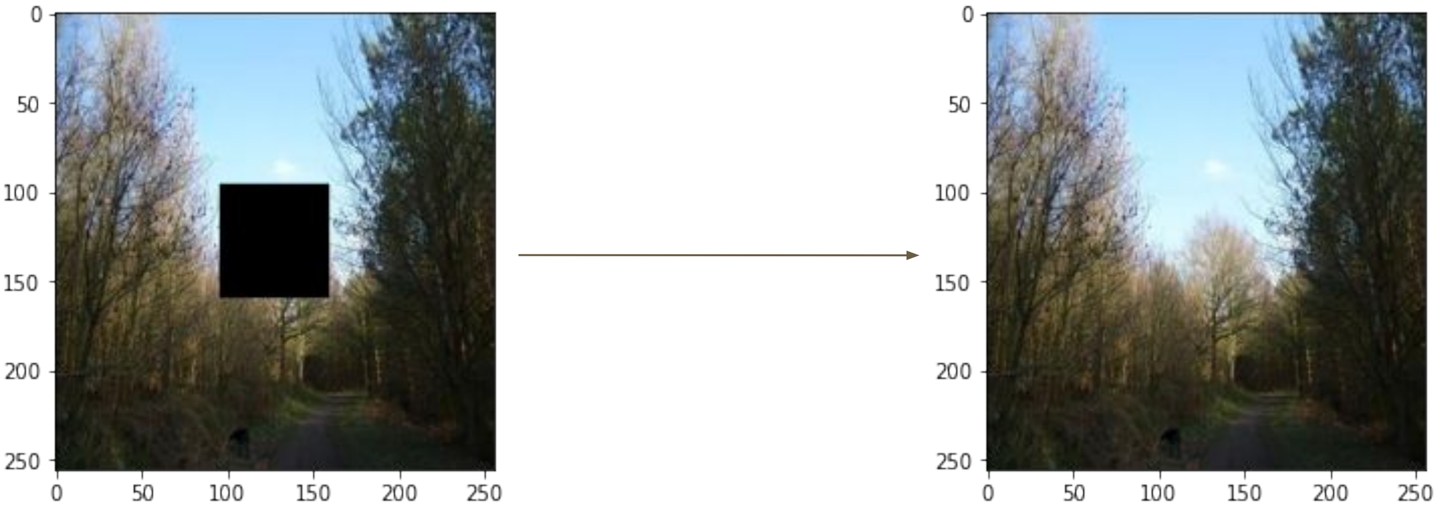
Context Encoder for Image Inpainting: Project Report

Samveed Desai
Electrical and Computer Engineering
A59005733

Sambaran Ghosal
Electrical and Computer Engineering
A59005052

Motivation

- Image inpainting: task of reconstructing damaged or missing parts of an image, so as to present a complete image



Main Idea

- Implement a pixel-prediction based approach: Context Encoder
- Develop convolutional neural networks that are used to generate particular image regions, based on the surroundings of the image
- Surroundings provide the necessary context to the model to learn better features to produce more realistic estimates for the missing image region

Previous Works

- Image Generation based
  - CNN's that can learn to generate novel images of certain object categories (chairs and faces)
  - Drawback:
    - Heavily rely on the labels corresponding to the input images

- Classical Inpainting and Hole-filling based
  - Using scene completion involving a cut-paste formulation using nearest neighbours from a large dataset
  - Drawback:
    - Used to fill in holes which were formed by removing only whole objects
    - Relies on a manually defined distance metric which is not transferable across different scenarios

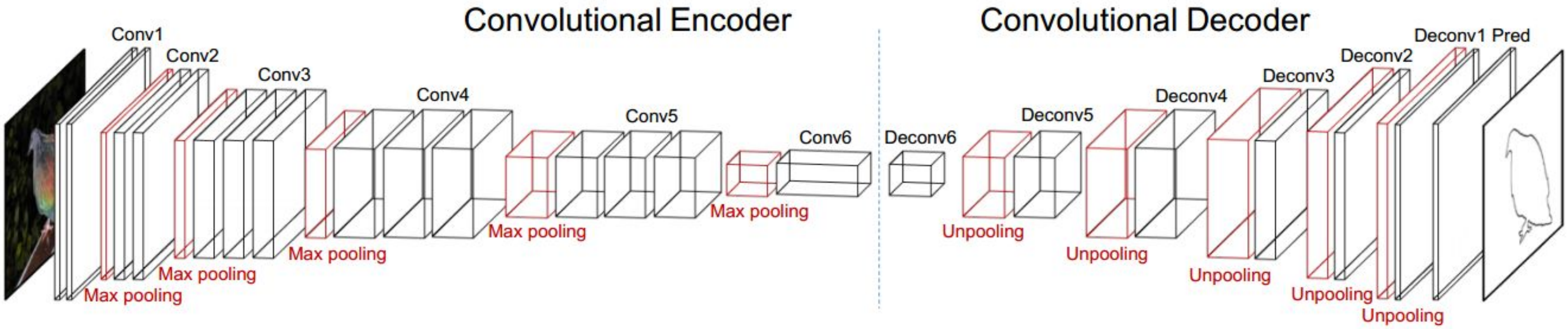
Our Method

- Use 2 main architectures for the Image Inpainting task:
    - Variational Autoencoders
    - Generative Adversarial Network

- Variational Autoencoders (VAE) project input image to a latent space distribution using encoder architecture and decoder reconstructs from samples of this distribution.

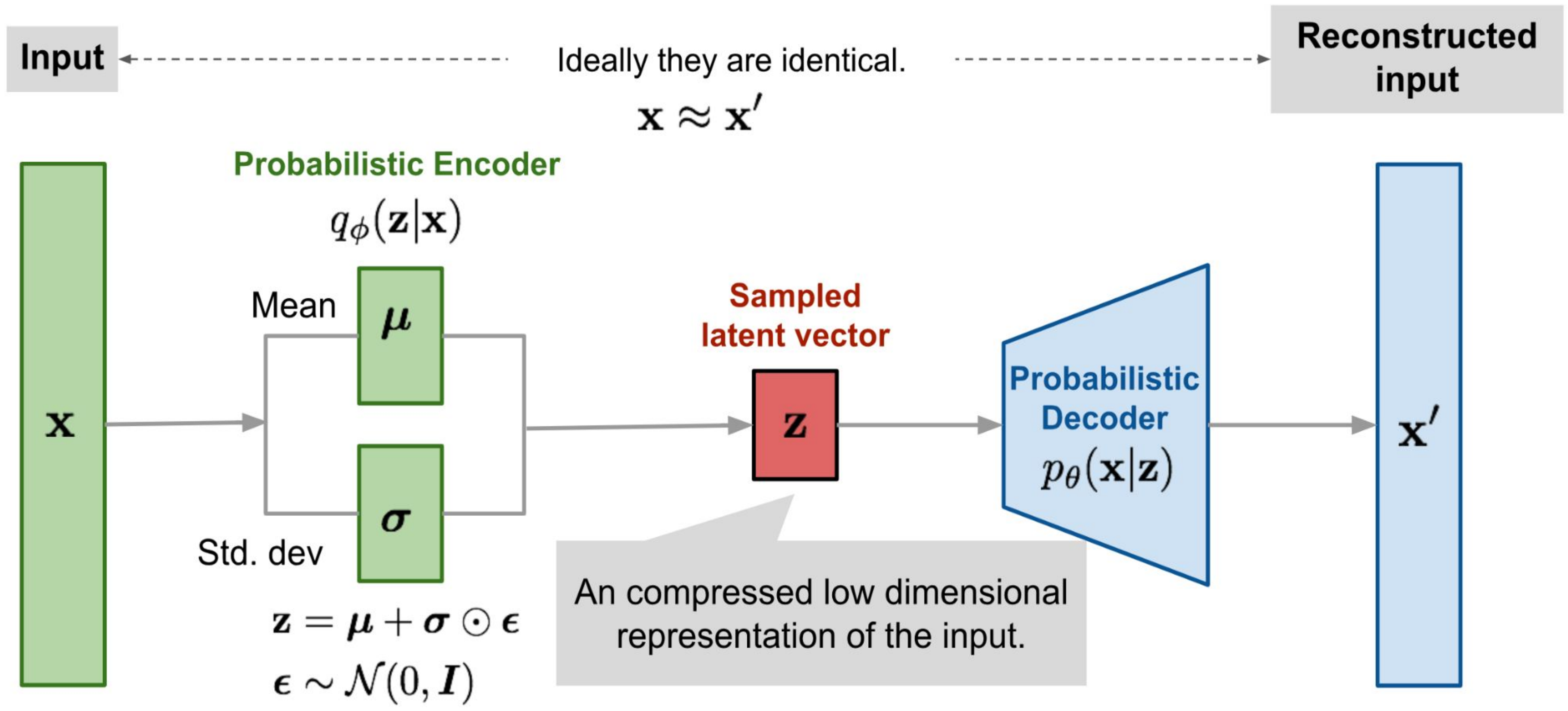
- Generative Adversarial Network (GAN) produces a reconstructed image using the encoder-decoder pipeline but an additional discriminator network tries to improve the quality of the generator.

Encoder-Decoder Pipeline



- Encoder learns a compact representation of the input image in a lower dimensional space.
- Decoder tries to reconstruct the input image from the latent representation learned by the encoder.

Variational Autoencoder



VAE Loss Function

The VAE loss function comprises of two terms : Reconstruction and KL-Divergence

$$\mathcal{L}(x) = ||x - D(z)||_2 + KL[\mathcal{N}(\mu, \Sigma)||\mathcal{N}(0, I)]$$

L2 reconstruction loss KL-Divergence Loss

KL-Divergence measures similarity between two different distributions

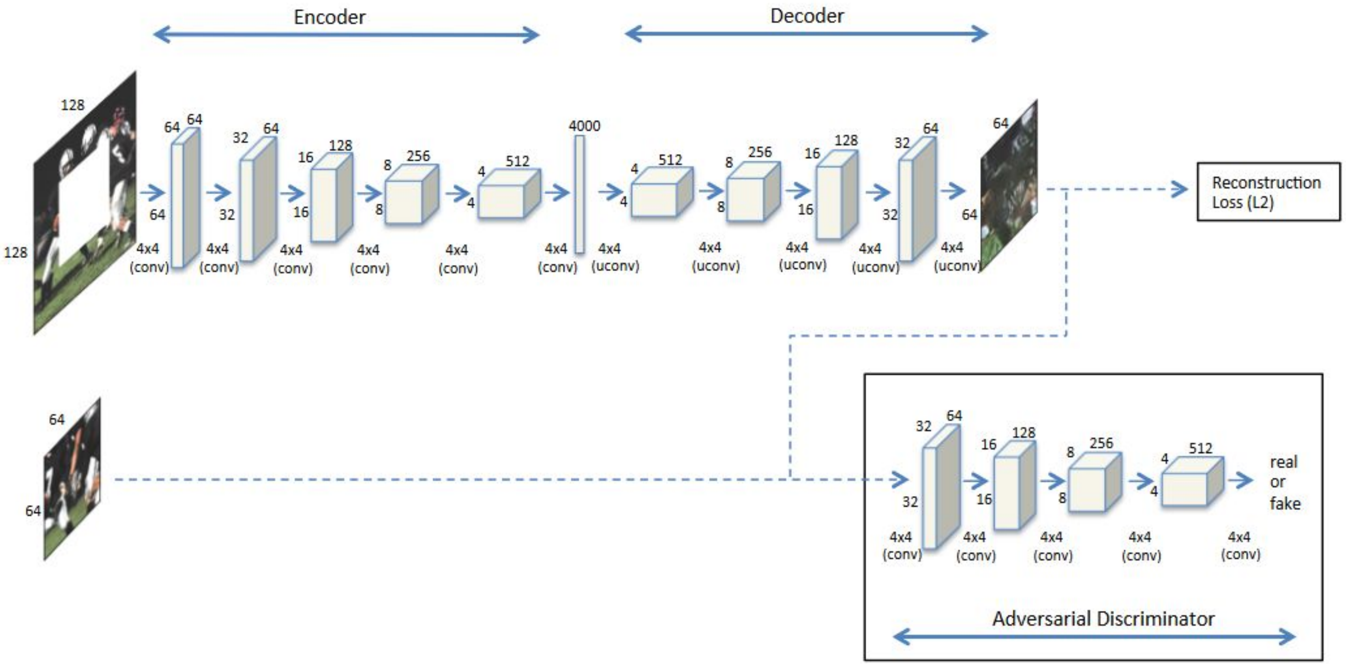
$$KL[p||q] = \int_{-\infty}^{\infty} p(x) log \frac{q(x)}{p(x)}$$

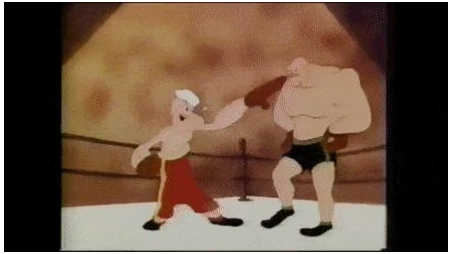
Encoder

- Derived from the AlexNet architecture

- Passes the input of $256 \times 256 \times 3$ through a series of convolutions, leaky rectified linear units (Leaky ReLU) activation functions and BatchNorm $\rightarrow$ $4 \times 4 \times 512$ dimensional feature representation

Generative Adversarial Networks





Discriminator vs Generator

GAN Loss

- Adversarial loss:

$$\mathcal{L}_{adv} = \max_{D} \mathbb{E}_{x \in \mathcal{X}}[\log(D(x)) + \log(1 - D(F((1 - \hat{M}) \odot x)))]$$

- Reconstruction Loss:

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2^2$$

- Total Loss:

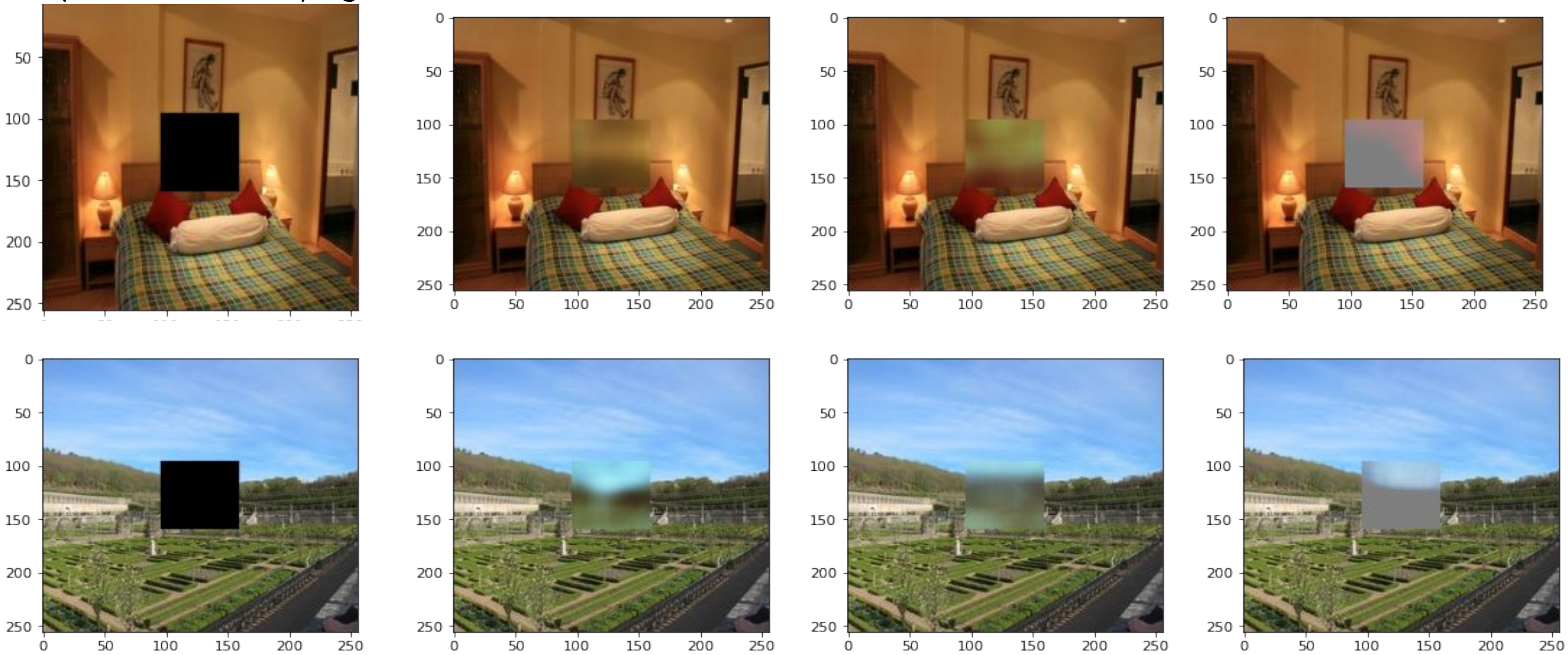
$$L_{total} = \lambda_{rec} L_{rec} + \lambda_{adv} L_{adv}$$

Experiment setup

- Places Dataset, 2.5 million images of over 205 scenes
- Input image: 256*256, with central 64 * 64 portion masked out
- Training Data: 5000 images, Test Data: 2000 images, Batch size: 32
- Number of epochs: 100, Bottleneck Number: 4000

Results

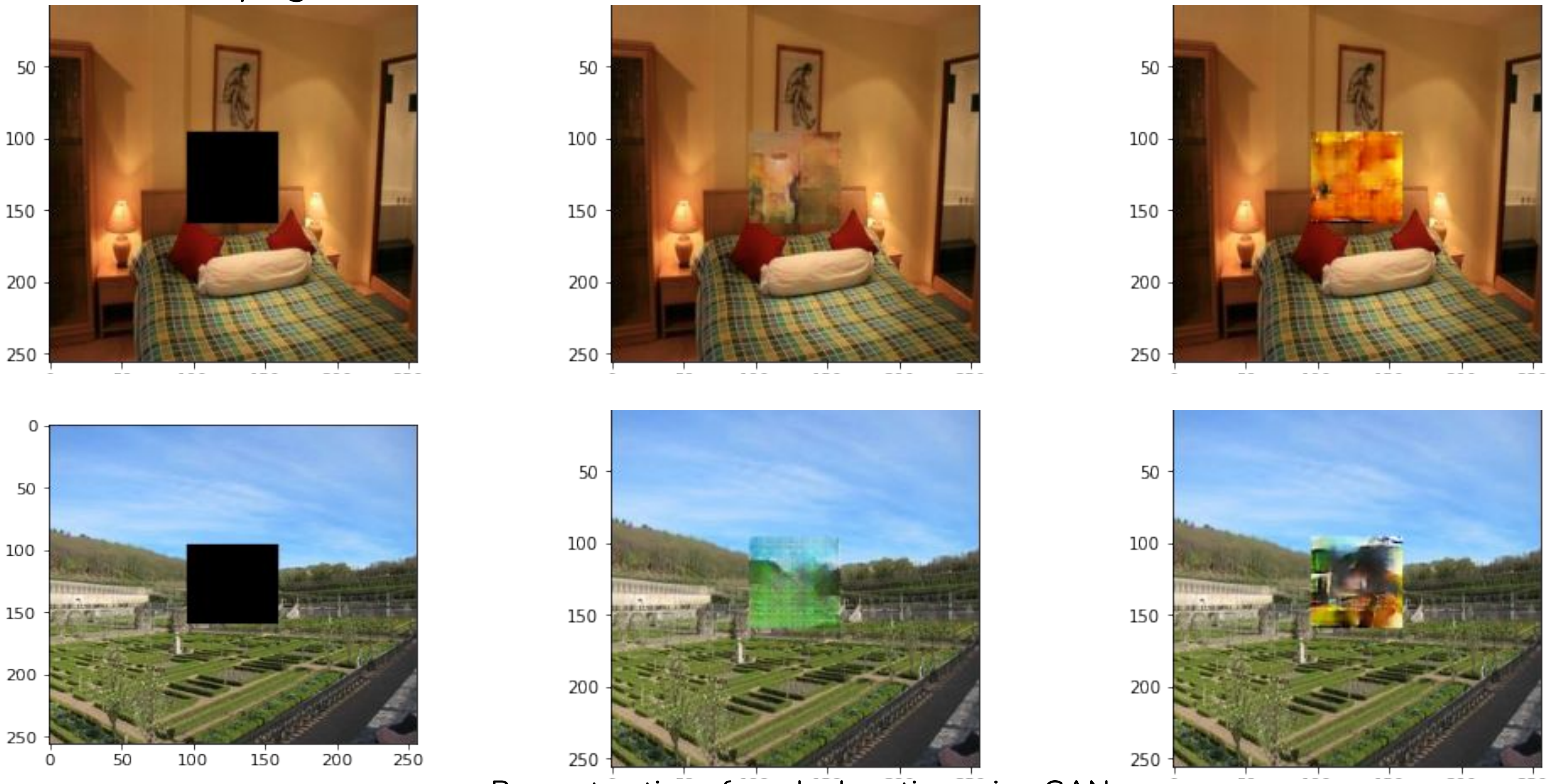
Experiment 1 : Varying encoder architectures in VAE



Reconstruction of masked portion using Variational Autoencoder.
Column1 : original, Column2 : VAE VGG, Column3 : VAE ResNet, Column 4 : VAE AlexNet

Results

Experiment 2 : Varying encoder architectures in GAN



Reconstruction of masked portion using GAN.
Column1 : original, Column2 : GAN AlexNet, Column3 :GAN ResNet

Quantitative Results

Table 1: Comparison between different encoder architectures in VAE

Network	Training Loss	Test Loss
AlexNet	1894.29	1882.63
VGG	1747.96	1784.43
ResNet	1725.11	1769.02

Table 2: Comparison between different encoder architectures in GAN

Network	Training Loss	Test Loss
AlexNet	1798.83	1768.69
ResNet	3030.98	3038.22

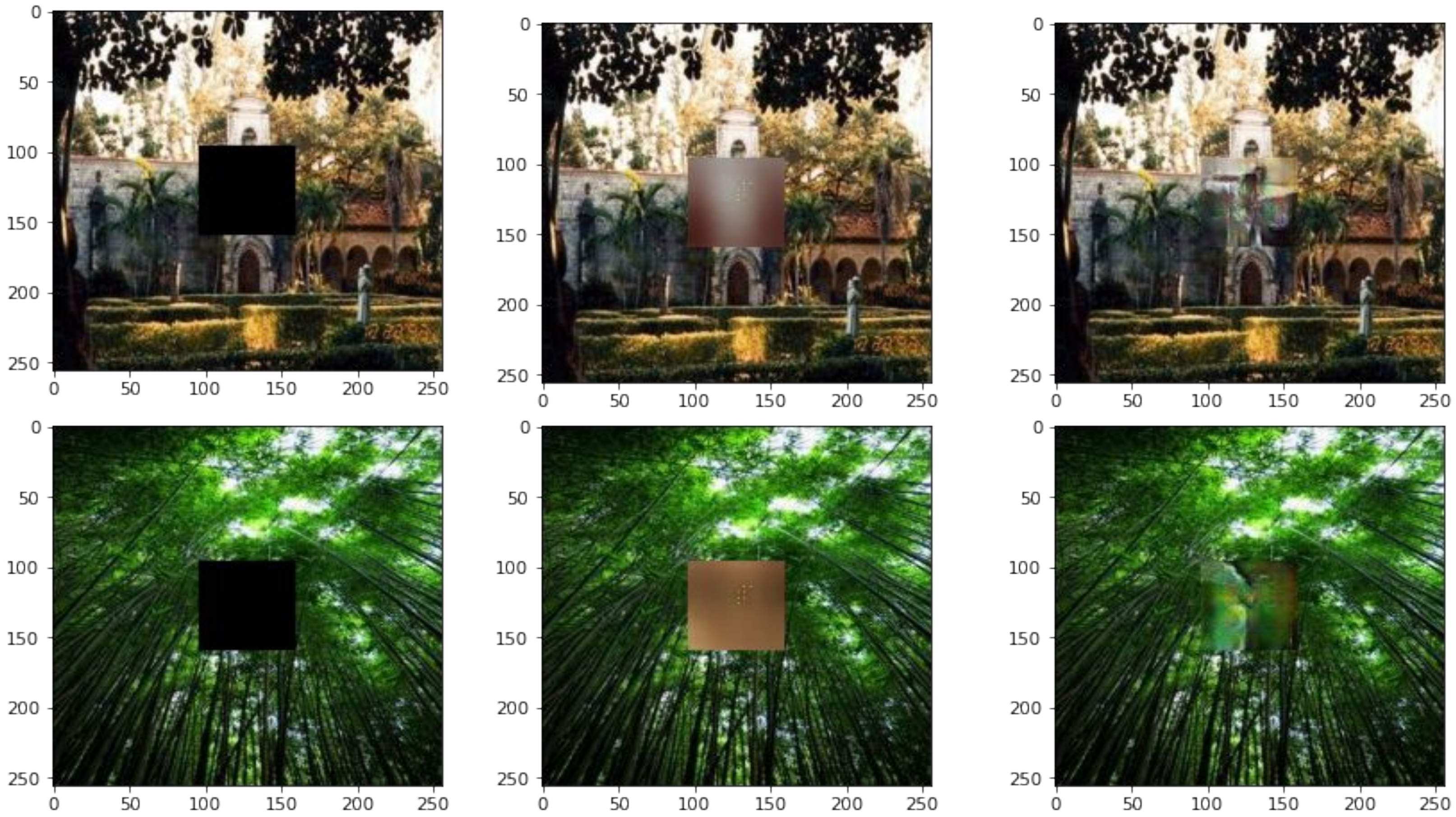
Note: Loss is evaluated on the basis of L2 reconstruction loss between the generated image and the original image, averaged over all the batches present in the training and test set

Experiment 3: Effect of bottleneck size on the performance

Table 3: Comparison between bottleneck size in reconstruction

	Bottleneck Size		
	500	2000	4000
VAE VGG	1817.82	1784.43	1808.84
VAE ResNet	1819.92	1813.5658	1818.61
GAN AlexNet	2101	2237.55	1813.56
GAN ResNet	5470.18	4704.27	3038.22

Who wins : VAE vs GAN



Conclusion

- Reconstructions from VAE were closer in resemblance to what was present in the original image but the reconstructions turned out to be super blur

- Reconstructions are very sharp and clear obtained from GAN’s although they may not exactly represent the original image pixels

- For reconstruction tasks, including a term for adversarial loss often improves the performance

References

1. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darell, A. Efros, Context Encoders: Feature Learning by Inpainting, CVPR 2016

2. J. Jordan, Variational autoencoders, 2018, https://www.jeremyjordan.me/variational-autoencoders/

3. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014