Efficient Latent Space Representation Learning

Himanshu Arora
Simon Fraser University

Raksha Harish
Simon Fraser University

Tavleen Sahota
Simon Fraser University

Shimon Smith
Simon Fraser University

Abstract

Generative models in deep learning have been used extensively to create new instances of data such as text and images. For a given set of data instances X, and labels Y, these models have the ability to either capture the joint probability p(X,Y), or just capture the individual probability p(X). The joint probability is useful in a supervised learning setup, where the data generation can be conditioned on the input labels, whereas individual probabilities are used in an unsupervised learning setup. In this project, we aim to develop a specific generative model, called the Variational Autoencoder (VAE) in order to learn depth (D) information along with RGB information for a given image dataset, and then use this information to successfully reconstruct RGB-D images from the latent vectors. VAEs are used in generation of image datasets, human faces, text to image translation, video sequencing and prediction, data compression, and vision-based robotic navigation.

1 Introduction

An Autoencoder is a type of artificial neural network, that can learn a specific representation of data, by generally reducing the dimension of the given data and training the network to learn the features. It consists of an encoder, a bottleneck layer, and a decoder. The layer between the encoder and decoder is known as the latent space. The autoencoder works by compressing the input into a latent space representation, and then reconstructing the output from this representation. In the process the latent space learns an encoded, compressed representation of an image, which is be used by the decoder to reconstruct the original image.

Variational Autoencoders (VAEs) [1] have a similar architecture to autoencoders but unlike autoencoders, VAEs are generative models that encode data in a latent space as regularized distributions, and then decode it to reconstruct the data. VAEs provide a probabilistic approach for learning latent space embeddings. Due to the compression in the latent space and probabilistic interpretation, the outputs generated by the decoder are usually degraded and blurred compared to the original inputs. Usually VAEs generate better results, when there is less variance observed in the training data, such as using a specific class of input images.

In the past, VAEs have been successfully trained to encode RGB image data, and reconstruct it. However, there is not much research for the ability of VAEs to encode and reconstruct the depth (D) information that is present in any given image. Our aim is to build a bespoke VAE model, that is capable of efficiently learning the depth (D) of an image, along with the RGB information, and then reconstructing the RGB-D images. We start by using RGB-D inputs to train our model to reconstruct RGB-D outputs. The final goal is to use only RGB images, and encode the depth information as well. A number of models were implemented to reconstruct RGB-D images, and these models have been compared to find the most optimal architecture for RGB-D reconstruction.

We show our results on two datasets:

1. MNIST Dataset of handwritten digits (labelled dataset) – 60,000 training samples and 10,000 test samples of RGB images

2. NYU Depth Dataset V2 – consists of a variety of indoor scenes recorded by both RGB and Depth cameras with 464 scenes, 407,024 unlabelled frames and 1449 processed pairs of RGB and depth images

2 Related Work

This section presents the different methods that have been reviewed for learning depth and RGB information from image data in order to generate the respective reconstructions. Comparison of the following methods and combination of certain techniques has helped in the development of a novel architecture for our work.

A high-quality dense depth map is estimated from a single RGB image in [2], wherein the distribution of depth values are modified using building blocks called "AdaBins". Since a single image is used as input, this model performs relatively better due to less variance in the input. A general encoder-decoder architecture is used to refine the latent space output, wherein convolutions are used in post-processing blocks, in order to retain the high resolution of the image.

The Monocular Depth Estimation method is used in [3], to learn the depth information along with RGB in the latent space. In this paper, the authors have reviewed several supervised, semi-supervised and unsupervised depth estimation methods, and presented the challenges in estimating the depth in an unsupervised setup. It has been concluded that monocular depth estimation works best when a supervised approach is implemented using a labelled dataset. Generative Adversarial networks tend to perform better than VAEs when deeper neural networks are implemented in an unsupervised setup. Compared to [3], the local planar model used in [4] performs better for estimating monocular depth information from 2D RGB images. This is because, novel local planar guidance layers are located in the decoding phase of the architecture, which reduces the degradation of the spatial resolution of the estimated outputs. Hence, down sampling is implemented in the latent space in to preserve the features in [4] guidance layers of the decoder and produce reconstructions of better quality/resolution.

In [5], the authors presents supervised deep learning techniques for latent space representation, and then evaluating the VAE performance for MNIST, CIFAR-10 and IVUS datasets. Adaptive sampling technique is used to adaptively select informative samples, and feed only the most relevant samples to the VAE in order to reconstruct the RBG image from the latent space using nearest neighbour, and interpolation-based sampling. This paper does not implement reconstruction of depth information as it evaluates the models only on RGB datasets.

An unsupervised learning framework is implemented in [6] to estimate monocular depth and camera motion from unstructured video sequences. KITTI dataset has been used for evaluating the framework, and it has been concluded that, monocular depth performs better for supervised models, wherein low-variance and comparable inputs are used. Unsupervised approach works well only when consistency is observed in both the inputs and the latent space representation.

3 Methodology

3.1 Comparative Models

We have implemented the following 3 models on MNIST and NYU Depth V2 datasets in order to study the changes observed in the outputs of VAE reconstructions. The models are:

1. Fully Connected (FC) VAE – consists of Encoder and Decoder with a sequential set of fully connected layers.

2. Conditional VAE (C-VAE) – This is an extension of the Vanilla VAE, where the image reconstruction is conditioned on the latent vector space to control the generation of data by the decoder based on the labelling used in MNIST dataset and the scenes present in NYU Depth V2 dataset.

3. VGG-16 based VAE – In this model, a VGG block containing 16 convolutional layers is defined and used by the encoder to improve the encoder's performance for compressing the original image.

It was observed that FC, C-VAE and VGG models performed very well on MNIST dataset as it contains only RGB image inputs and there is no need to reconstruct depths for MNIST dataset. The same 3 models performed poorly on NYU Depth V2 dataset, as it is an unlabelled dataset wherein the depth information needs to be reconstructed along with RGB information. Hence, an unsupervised approach is considered while building a bespoke architecture for NYU Depth V2 reconstruction, to efficiently learn and reconstruct both depth (D) and RGB information from the latent space.

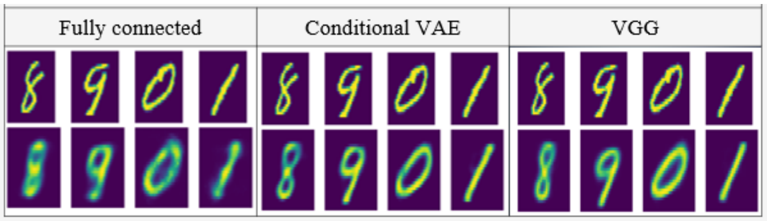

Figure 1: Ground truth RGB images (top rows) and reconstruction (below) for FC, C-VAE and VGG models for MNIST dataset at epoch 20

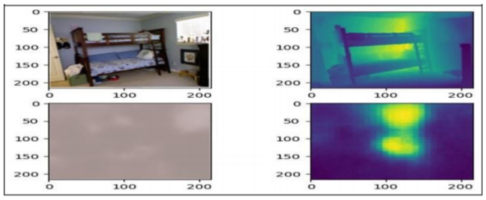

Figure 2: Ground truth RGB and Depth images (top rows) and reconstruction (below) for VGG model for NYU Depth V2 dataset at epoch 100

3.2 Proposed Model and Loss Formulation

In our proposed model we aim to learn a latent space that encodes both RGB and depth of scene images. As a starting point, we present a VAE that takes RGB-D images and encodes them in a compact latent space and reconstructs RGB-D images as well. Afterwards, we remove the dependency on depth at test time by means of monocular depth estimation. In this section we present the probabilistic model and a convolutional VAE architecture, followed by a discussion of training details and the dataset used.

In our VAE we use a Gaussian likelihood. The mean is parameterized by our decoder network, μ_θ, and the variance is taken to be 1. The input RGB-D images are assumed to be i.i.d over spatial dimensions and the 4 channels. Thus the likelihood decomposes as a product of single variable Gaussians.

$$p(x|z) = \prod_x \mathcal{N}(\mu_\theta(z), 1) \tag{1}$$

As a prior over the latent variable, z, we take an isotropic Gaussian with mean 0 and standard deviation 1.

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{1}) \tag{2}$$

Lastly our variational approximation of the posterior $p(\mathbf{z}|\mathbf{x})$ is also Gaussian. The mean and variance are both parameterized by the encoder network. We denote them, μ_ϕ and σ_ϕ.

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{z}), \sigma_\phi(\mathbf{z})) \tag{3}$$

To train our VAE we maximize the ELBO. As in most works that train VAEs, we use the formulation of the ELBO as the sum of an expected log likelihood reconstruction term and a KL regularization

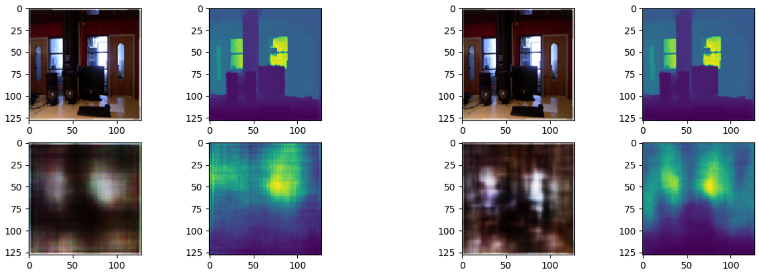

Figure 3: At epoch 10 (left) the reconstruction lacks detail in both the RGB and depth reconstruction. After training for 200 epochs (right) the reconstruction includes more detail in NYU Depth V2.

term. As a result of both our prior and posterior approximation being Gaussian, the KL term can be integrated analytically and computed efficiently [1].

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})\,||\,p(\mathbf{z})\right)\\ &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z})\right] + \frac{1}{2}\sum_{j=1}^{J}\left(1+\log(\sigma_j^2)-\mu_j^2-\sigma_j^2\right)\end{aligned}$$

Here J is the latent dimension and μ, σ are the variational parameters predicted from x by the encoder.

3.3 Architecture

We adopt a convolutional architecture for both our encoder and decoder, modeled after the work of Hou, et al [7]. Our encoder downsamples the input by using strided convolutions. The encoder consists of 5 convolutional layers each with a kernel size of 4×4, 1 pixel of zero-padding and a stride of 2. Each layer is followed by batch normalization and a leaky relu activation with a slope of 0.2. The depth in each of these layers is as follows: 64, 128, 256, 512, 512. The convolutional layers are then followed by a fully connected layer to compute the mean and variance of the latent variable. In our experiments we use a latent dimension of 400. For the decoder we begin with a fully connected layer that maps the latent dimension to a 4×4 tensor input to the convolutions. We found that upsampling using nearest neighbour interpolation followed by a convolution produced visually better results than transpose convolutions. The nearest neighbour interpolation layers are followed by 3×3 convolutions. We followed [7] and used replication padding for these convolutions. All but the last of these layers have batch normalization and leaky relu activations. The depths are decreased across 5 such layers symmetrically to the encoder. Afterwards we apply a $\tanh$ activation to map the predicted mean to the $[-1, 1]$ range. Our inputs are also normalized into this range.

3.4 Training

Our model is trained for 200 epochs with a learning rate of 1e-4 with Adam [8]. Figure 3 shows how our VAE gradually learns to encode and decode details of the scene to and from the latent space. Early in training we see a blurry reconstruction and later details such as distinct panels on the window start to be visible. In figure 4 we visualize the two components, KL and log likelihood, of the ELBO loss, followed by the ELBO and MSE computed over the test set. We can see that the ELBO starts to decrease and the MSE increases around epoch 80. This is often a sign of overfitting but in our case we found that valuable detail was still being added to the reconstructions.

We train our model on the NYU Depth dataset [9]. The dataset contains RGB-D images from 464 different indoor scenes. The images are collected as video sequences from a kinect camera. 1449 images come with the depth preprocessed from the raw depth values. The processed depth values are computed by projecting raw depth data into the coordinate space of the RGB image to align the two images. Once aligned there may be missing values in the depth image which are inpainted using the colourization method of Levin, et al [10]. Ma, et al [11] provide a larger preprocessed subset of NYU Depth where they inpaint the projected depth values using a cross-bilateral filter. We found that using this larger set of processed data did not improve results. More so, there was some artifacts in many of

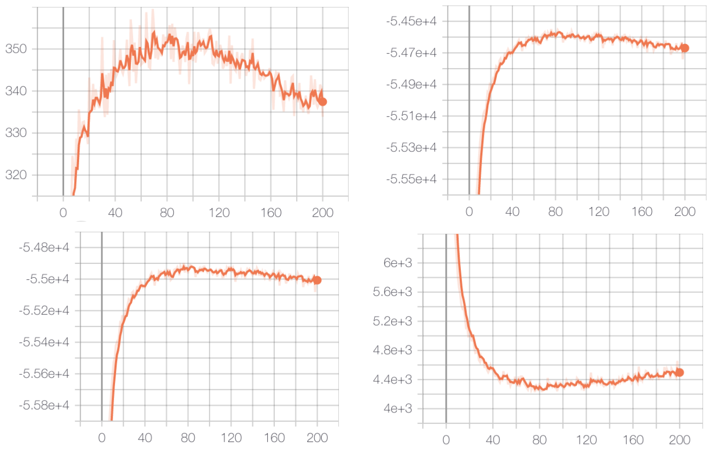

Figure 4: KL (top left), likelihood (top right), ELBO (bottom left) and MSE (bottom right) computed over the test set during training for NYU Depth V2 dataset

their depth images, so if more data was to be used for training, we would apply [10] on the raw data instead. Following other works, we used the official train/test split in which 249 scenes are taken for training and 215 for testing. All images are resized to 128 × 128 and normalized to values in the range $[-1, 1]$ for our network.

3.5 Results

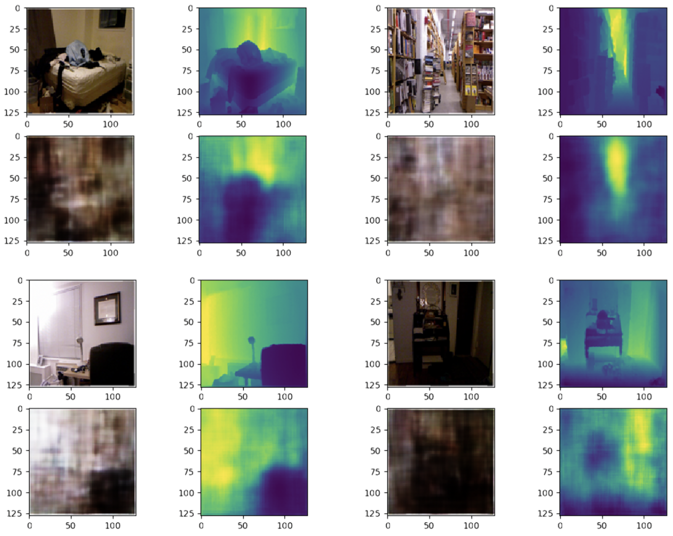

Figure 5: Ground truth RGB and depth image pairs (top rows) and reconstruction (below) for NYU Depth V2 dataset

We use MSE as the primary metric for evaluating our architecture, as we aim for high quality reconstructions given a compact latent representation. On our test set the model achieves an MSE of 4464 after 200 epochs. The distribution over RGB-D images of indoor scenes is complex, as each scene has unique details. Even with a deep convolutional architecture it is difficult to achieve high quality reconstructions that capture detail. It is known that VAE reconstructions are blurry and methods have been proposed to add detail into VAE reconstructions. This includes using a perceptual loss such as in [7], where the loss is constructed from features extracted from a pretrained CNN. These deep features can be used to get a better measure of similarity between the input and reconstruction produced by the VAE yielding higher quality reconstructions. [12] uses a learnable

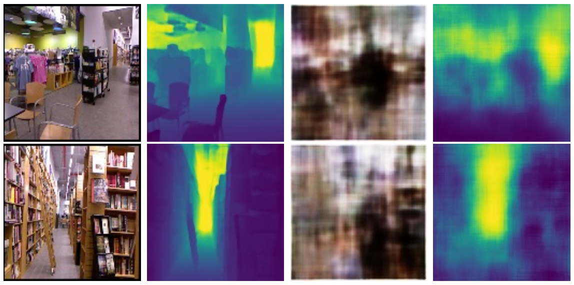

Figure 6: From left to right, ground truth RGB, monocular depth estimation via BTS [4], our VAE RGB and depth reconstructions

prior (PixelCNN) in addition to a discrete latent representation. In our work the goal was to use a plain VAE since we aim for real time inference in a reinforcement learning environment with limited hardware resources. Figure 5 shows our reconstructions on 4 RGB-D image pairs from the test set. We can see that objects are reconstructed both in the RGB and depth images correctly.

4 Depth Estimation from RGB images

Next, we aimed to remove the dependency on depth at test time by means of monocular depth estimation. After a thorough literature review, we implemented a recently published method of monocular depth estimation using multi-scale local planar guidance [4] to feed depth information to our proposed VAE model. This enabled our proposed model to estimate depth from a single RGB image (which demands less cost than using a sequence of images).

The network architecture proposed by Lee et al. uses novel local planar guidance layers located at multiple stages in the decoding phase and gives outstanding performance compared to competing approaches. It uses an encoding-decoding scheme that first reduces the feature map resolution to 1/8th of the original resolution and then recovers back to original resolution for dense prediction. A backbone network is used for dense feature extraction and a denser atrous spatial pyramid pooling layer is used to extract contextual information. Then the local planar guidance (LPG) layer is used at each decoding phase stage for geometric guidance to the target estimated depth. Finally, outputs from all proposed layers are concatenated and fed to the final convolution layer for depth estimation [4].

A major advantage of this architecture is that it doesn't use simple nearest neighbor unsampling layers to recover back to the original resolution and skip connections from encoding stages. Instead, it uses LPG layers to recover to full resolution and use them together to get the estimated depth [4].

We used the pre-trained weights from the Lee et al. implementation to obtain depth estimation from an input RGB image. Then the RGB image and depth estimation were reconstructed using the pre-trained proposed VAE weights. In figure 6 we show our RGB and depth reconstructions when using the depth estimation as input in place of ground truth depth.

5 Future Work

We hope that our method will be applicable in a reinforcement learning setting where real time inference with limited hardware resources are key. As future work we hope to develop a single network that incorporates key features of this model and reconstructs RGB-D data from RGB inputs more efficiently.

6 Conclusion

In this project, we studied and implemented FC, C-VAE and VGG models for both MNIST and NYU Depth V2 datasets. The comparision of RGB-D reconstruction results led us to build our proposed model to efficiently encode depth information along with RGB, and reconstruct the same from the latent space with reduced error. From this experiment, we found that VAEs give better reconstruction results for labelled datasets like MNIST. Hence, deeper neural networks must be used in the encoder and decoder to efficiently learn and reconstruct RGB-D information in an unsupervised setup, wherein unlabelled datasets like NYU Depth V2 are used. We also found that consistency in the input data leads to better reconstructions using VAEs. After building a custom VAE architecture we removed the dependency on depth at test time by employing the state of the art in monocular depth estimation. This project can be further extended for vision-based robotic navigation and also in self-driving cars, where the depth information must be present along with RGB information to guide the agent to navigate smoothly and avoid any accidents.

7 Contributions

Himanshu Arora - Dataset, Literature Review, Proposed VAE Implementation in Google Colab, Debugging, Poster Preparation, Report Writing

Raksha Harish - Dataset, Literature Review, Comparative Models Implementation in Google Colab, Debugging, Poster Preparation, Report Writing

Tavleen Sahota - Dataset, Literature Review, Depth Estimation from RGB images Implementation in Google Colab, Debugging, Poster Preparation, Report Writing

Shimon Smith - Dataset, Literature Review, Proposed VAE Implementation in Google Colab, Debugging, Poster Preparation, Report Writing

8 GitHub Repository

Link : `https://github.com/shimismith/efficient-latent-representation`

Branch : "main"

Tag : "final"

References

[1] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2014.

[2] S. F. Bhat, I. Alhashim, and P. Wonka, “Adabins: Depth estimation using adaptive bins,” 2020.

[3] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, “Monocular depth estimation based on deep learning: An overview,” *Science China Technological Sciences*, vol. 63, p. 1612–1627, Jun 2020.

[4] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, “From big to small: Multi-scale local planar guidance for monocular depth estimation,” 2020.

[5] Y. Mo, S. Wang, C. Dai, R. Zhou, Z. Teng, W. Bai, and Y. Guo, “Efficient deep representation learning by adaptive latent space sampling,” 2020.

[6] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” 2017.

[7] X. Hou, L. Shen, K. Sun, and G. Qiu, “Deep feature consistent variational autoencoder,” 2016.

[8] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.

[9] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.

[10] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” in *ACM SIGGRAPH 2004 Papers*, SIGGRAPH ’04, (New York, NY, USA), p. 689–694, Association for Computing Machinery, 2004.

[11] F. Ma and S. Karaman, “Sparse-to-dense: Depth prediction from sparse depth samples and a single image,” *CoRR*, vol. abs/1709.07492, 2017.

[12] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” *CoRR*, vol. abs/1711.00937, 2017.

[13] P. Chakravarty, P. Narayanan, and T. Roussel, “Gen-slam: Generative modeling for monocular simultaneous localization and mapping,” 2019.

[14] S. Levine, “Reinforcement learning and control as probabilistic inference: Tutorial and review,” 2018.

[15] C. Doersch, “Tutorial on variational autoencoders,” 2016.