Disentangling Latent Factors of Variation for Visual Data

Divyanshu Talwar
Roll Number: 2015028

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering
on December 31, 2018

BTP Track: Research Track

BTP Advisor : Dr. Saket Anand

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **"Disentangling Latent Factors of Variation for Visual Data"** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under the guidance of **Dr. Saket Anand**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

..... **Place & Date: .....**
Divyanshu Talwar

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

..... **Place & Date: .....**
Dr. Saket Anand

Abstract

Disentangling higher level generative factors as disjoint latent dimensions offer several benefits such as ease of deriving invariant representations, targeted data augmentation with style-transfer, better interpretability of the data, etc. In this work, we focus on disentangling factors of variation with weak-supervision (in the form of pair-wise similarity labels) using a non-adversarial approach. We show compelling results for both the quality of disentangled representations and image generation for MNIST and CMU MultiPIE datasets, and UTK-face and CelebA datasets for cross-dataset evaluation. We further demonstrate few-shot learning of new previously-unseen classes as a consequence of effective disentangling of the latent subspace (into style and class).

Keywords: Disentangling Factors of Variation, Generative Adversarial Networks, Cycle-Consistent Architecture, Auto-encoders, Few-shot learning.

Contents

Chapter 1

Introduction

Machine learning algorithms perform better when supplied with effective feature representations of the raw data. Nowadays we see extensive use of deep learning models owing to their capability of producing (generally) better results. When looked upon with a different perspective, the working of a deep learning model can be interpreted in a way where the initial layers supply a suitable representation to their successors for successfully performing the task at hand. This, strengthens the need for producing effective representations of raw data. The effectiveness of a representation depends upon how well it captures the underlying latent factors that are relevant for the end task while ignoring the inconsequential or nuisance factors. The disentangling branch of representation learning paradigm focuses on producing disjoint subsets of the latent space which can later be used for various tasks.

Learning disentangled representations offer several advantages, such as (i) ease of deriving invariant representations which help in making the learned representations invariant of the factors inconsequential to the task at hand (for instance - pose, expression and illumination-setting invariant facial recognition); (ii) targeted data augmentation with style-transfer for producing more labeled-data to train upon; and (iii) few-shot learning ability - with the class and style separated it is now easier and to learn a new previously-unseen class with only a few samples at hand.

In this work, we focus on disentangling factors of variation with weak-supervision (in the form of pair-wise similarity labels) using a non-adversarial cycle-consistent variational autoencoder based approach [3]. We show compelling results for both the quality of disentangled representations and image generation on MNIST and CMU MultiPIE datasets, and UTK-face and CelebA dataset for cross-dataset evaluation. We further demonstrate few-shot learning of new previously-unseen classes as a consequence of effective disentangling of the latent subspace (into style and class).

The experiments section is divided into three subsections presenting the results based of quality of generated images and disentangled representations in the first two sections, and presenting the experiments with which we gained valuable insight regarding the problem and our model's performance. The quality of generated images section talks about our model competence when compared with Mathieu et al.'s [1] and Szabó et al.'s [2] work on the grounds of style-transfer

renderings, and demonstrates the effectiveness of learned disentangled subspaces using our model through compelling interpolation in the latent space, random sampling of the style space, and few-shot learning of previously unseen classes results. The next sub-section quantifies the purity of the disentangled subspaces and demonstrates our model's robustness to dimensionality change as compared to others. Finally, we discuss the results of DR-GAN by Tran et al. [21] for face frontalization and our efforts to improvement of image reconstruction quality experiments which lead to some valuable insights regarding the problem at hand and our model's working.

Chapter 2

Literature Survey

`Auto-encoders :`

Auto-encoders [4] [5] learn a latent-representation mapping of the input by minimizing the reconstruction error. The dimensionality of the latent space learnt can be either more, for sparse auto-encoders (weight normalization is done to prevent the degenerate solution) or less, for bottleneck auto-encoders, than the input dimension.

An auto-encoder consists of an encoder E and a decoder D. The encoder first maps the input data X into the latent space $H = \phi(E\{X\})$, where ϕ is a non-linear activation function. The decoder then maps this latent representation back to the input space $\tilde{X} = D\{H\}$. During the training phase, the encoder E and the decoder D are (usually) learnt by minimizing the following loss function :

$$\arg\min_{D,E} ||\tilde{X} - X||_F^2 \tag{2.1}$$

`Variational Auto-encoder (VAE) :`

Kingma et al. [6] present a variational inference approach for the auto-encoder based latent factor model. Consider the dataset $X = \{x_i\}$, $\forall i \in \{1, |X|\}$, where, x_i's are i.i.d samples each associated with a continuous latent variable z_i sampled from some prior $p(z)$ (generally, a variant of the standard normal distribution). The encoder, in the auto-encoder model, approximates the posterior term $q_\phi(z|x)$ and the decoder approximates the likelihood term $p_\theta(x|z)$ here, with ϕ and θ being the weights of the encoder and the decoder respectively. The encoder and the decoder are learnt by optimizing the following variational lower-bound :

$$\mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x)||p(z)) \tag{2.2}$$

$$where, KL(p(x)||q(x)) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \tag{2.3}$$

Where the loss function comprises of the expected value of the data likelihood and the KL-divergence term - which forces the learnt approximate posterior to align with the prior distribu-

tion of the latent space $p(z)$. Since sampling is not backpropagate-able, the authors use a linear transformation based reparameterization to enable the end-to-end training of the model. When learnt well (sufficiently small KL-divergence between the approximate posterior and the prior) VAE's can be used to generate data belonging to a favourable class (on which it was learnt on) by sampling from the prior $p(z)$ and passing this through the decoder.

`Generative Adversarial Networks(GANs) :`
GANs by Goodfellow et al. [7] have been shown to model complex, high dimensional data distributions and generate nice results from it. They comprise of two competing neural networks, trained together in an adversarial setting by optimizing the following loss function :

$$\max_G \min_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z))] \tag{2.4}$$

The discriminator D outputs the probability that a given sample belongs to input data distribution as opposed to being a sample from the generator generated distribution. On the other hand, the generator G is trained to map random samples from a prior distribution (usually, a variant of the standard normal distribution) in the latent space to samples from the true distribution. The training is said to have converged when the discriminator outputs $\frac{1}{2}$ for all generated samples. DCGANs [8] use CNNs as a part of the GAN architecture to generate samples from complex image distributions.

Despite of their ability generate compelling results, training GANs is quite tricky and requires carefully designed tricks [9].

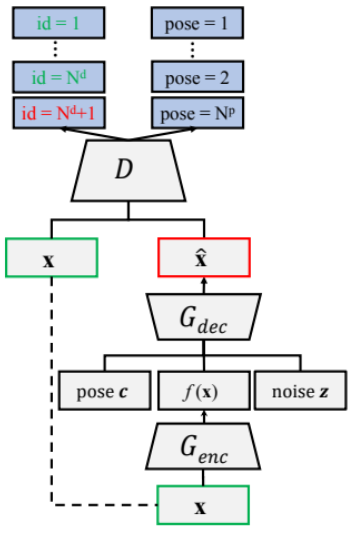


Figure 2.1: DR-GAN model

`Adversarial Auto-encoders(AAE) :`
Inspired by the idea of variational autoencoders, adversarial auto-encoders [10] use adversarial training instead of the standard KL-divergence loss to align the approximate posterior (learnt by the encoder) with an arbitrary prior distribution over the latent space variables.

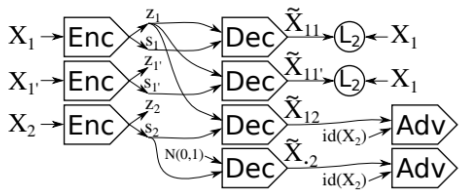


Figure 2.2: Mathieu et al. model training architecture. The inputs x_1 and x_1' are two different samples with the same label, whereas x_2 can have any label.

The model is trained in two-phases : the reconstruction phase, and the regularization phase. The reconstruction phase, is a standard auto-encoder branch where in the encoder and the decoder are learnt by minimizing the reconstruction error. Where as, the regularization phase forces the encoder to learn an approximate posterior which aligns with the prior distribution over the latent space variables by minimizing the adversarial cost of distinguishing between the positive samples belonging to the posterior form the negative samples belonging to the arbitrary prior.

`Conditional - GANS (CGANs) :`
CGANs [11] propose a way to generate data using GANs conditioned on some information y. This just amends the cost function of GANs s.t. the data and the prior are conditioned on the an auxiliary information random variable y.

$$\max_G \min_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x|y)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y))] \tag{2.5}$$

`SSGANS :`
SSGANs [12] change the discriminator of the GANs to now output the class labels too, i.e. for a dataset with N class labels, the discriminator D needs to predict one of the $N+1$ classes (N authentic/legible classes, plus, one fake/ill-legible class). The suggested change is implemented by replacing sigmoid with softmax as the activation function for the last layer of the network.

`PixelVAE :`
PixelVAE [25] is a VAE model with an autoregressive decoder based on PixelCNN. The model requires very few expensive autoregressive layers compared to PixelCNN and learns latent codes that are more compressed than a standard VAE while still capturing most non-trivial structure. Essentially, the model implements a tractable likelihood function (unlike the approximate estimated-lower-bound (ELBO) likelihood function used by the vanilla VAE) with the help of an autoregressive decoder.

`Disentangling factors of variation :` Initial works like [13] use the expectation-maximization (EM) framework to detect independent factors of variation which describe the input data. Tenenbaum et al. [14] try to solve the problem by learning bilinear maps from unspecified and specified parameters to images. In recent works, [15] [16] [17] Restricted Boltzmann Machines have been used to map factors of variation in images separately. Kulkarni et al. [18] model this problem as an inverse graphics problem and propose a network that disentangles transformation and lighting variations. In [19] and [20], invariant representations are learnt by removing the uninformative variables for a given task.

DR-GAN, by Tran et al. [21] using both identity and pose labels, claim to disentangle facial identity from pose. The architecture used is very similar to the the architectures of SSGANS, CGANS and AAEs put together. Where in they take an input image, pass it through the encoder to produce a feature map. This generated feature map along with the target pose code and random noise is fed as input to the decoder, which generates a face with same identity as of the subject in the input image in the target pose.
This model doesn't enforce disentangling of pose from the identity information, rather it adversarially learns a decoder, which selects the identity-specific information from the feature maps and mixes it with the target pose information in a non-linear fashion to generate an image projected in the target pose.

Disentangling approaches, like those of Szabó et al. [2] and Mathieu et. al [1] (model - 2.2) combine auto-encoders with adversarial training to disentangle specified from unspecified features of variation based on a single factor of variation's class labels. On the other hand, in Cyle-Consistent VAEs [3], Jha et al. have leveraged the idea of cycle-consistency in the unspecified latent space and to introduce a non-adversarial approach to disentangling factors of variation problem (model - 3.1).

Chapter 3

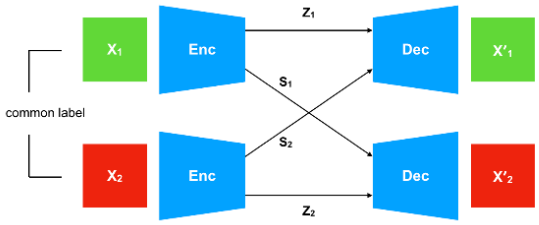
Methodology

In this work we extend the model proposed by Jha et. al [3] i.e. a conditional variational auto-encoder based model, where the latent space is partitioned into two *complementary* subspaces: s, which controls specified factors of variation associated with the available supervision in the dataset, and z, which models the remaining unspecified factors of variation. Similar to Mathieu et al.s [1] work Jha et. al keep s as a real valued vector space and z is assumed to have a standard normal prior distribution $p(z) = \mathcal{N}(0, I)$. Such an architecture enables explicit control in the specified subspace, while permitting random sampling from the unspecified subspace. Marginal independence between z and s is assumed, which implies complete disentanglement between the factors of variation associated with the two latent subspaces.

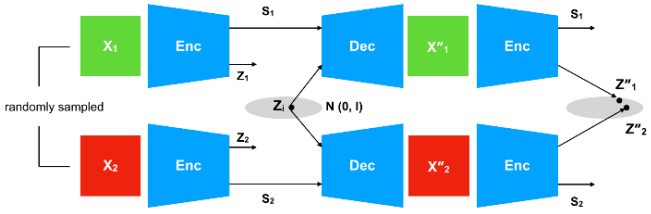
Encoder. The encoder can be written as a mapping $Enc(x) = (f_z(x),\ f_s(x))$ where $f_z(x) = (\mu,\ \sigma)$ and $f_s(x) =\ s$. Function $f_s(x)$ is a standard encoder with real valued vector latent space and $f_z(x)$ is an encoder whose vector outputs parameterize the approximate posterior $q_\phi(z|x)$. Since the same set of features extracted from x be used to create mappings to z and s, it is modelled using a single encoder with shared weights for all but the last layer, which branches out to give outputs of the two functions $f_z(x)$ and $f_s(x)$.

Decoder. The decoder, $x' = Dec(z,\ s)$, in this VAE is represented by the conditional likelihood $p_\theta(x|z, s)$. Maximizing the expectation of this likelihood w.r.t the approximate posterior and s is equivalent to minimizing the squared reconstruction error.

Forward cycle. A pair of images, x_1 and x_2, is sampled from the dataset that have the same class label. Then both of these images are passed through the encoder to generate the corresponding latent representations $Enc(x_1) = (z_1,\ s_1)$ and $Enc(x_2) = (z_2,\ s_2)$. The input to the decoder is constructed by swapping the specified latent variables of the two images. This produces the following reconstructions: $x'_1 = Dec(z_1,\ s_1)$ and $x'_2 = Dec(z_2,\ s_2)$. Since both these images share class labels, swapping the specified latent variables should have no effect on the reconstruction loss function. Thus, the conditional likelihood of the decoder can be re-written as $p_\theta(x|z, s^*)$, where $s^* = f_s(x^*)$ and x^* is any image with the same class label as x. Fig. 3.1 (a) shows a diagrammatic representation of the forward cycle.



(a) Forward Cycle



(b) Reverse Cycle

Figure 3.1: Jha et al. Model. (a) Forward Cycle : Image reconstruction done using VAEs by swapping the s latent variable between two images from the same class. The process works with pairwise identical labels, thus semi-supervised, - as the actual class labels of the sampled image pair are not required. (b) Reverse cycle of the cycle-consistent VAE architecture where in a point sampled from the z latent space is combined with class specific factors from two separate sources, generating two different images. However, on passing these generated images through the encoder again, we should be able to obtain the same sampled point in the z space.

$$\min_{Enc,\ Dec} \mathcal{L}_{forward} = -\mathbb{E}_{q_\phi(z|x,s^*)}[log\ p_\theta(x|z,s^*)] + KL(q_\phi(z|x,s^*) \parallel p(z)) \tag{3.1}$$

It is worth noting that forward cycle does not demand actual class labels at any given time. This results in the requirement of a weaker form of supervision in which images need to be annotated with pairwise similarity labels. This is in contrast with the previous works of Mathieu et al. [1], which requires actual class labels, and Szabo et al. [2], which requires image triplets.

Reverse Cycle. The reverse cycle shown in Fig. 3.1 (b) is designed based on the idea of cycle-consistency in the unspecified latent space. z_i is sampled from the Gaussian prior $p(z) = N(0, I)$ over the unspecified latent space and is passed through the decoder in combination with specified latent variables $s_1 = f_s(x_1)$ and $s_2 = f_s(x_2)$ to obtain reconstructions $x_1'' = Dec(z_i, f_s(x_1))$ and $x_2'' = Dec(z_i, f_s(x_2))$ respectively. Here it is preferable that both x_1 and x_2 not have the same labels. Since both images x_1'' and x_2'' are generated using the same z_i, their corresponding unspecified latent embeddings $z_1'' = f_z(x_1'')$ and $z_2'' = f_z(x_2'')$ should be mapped close to each other, regardless of their specified factors. Such a constraint promotes marginal independence between z and s, as images generated using different specified factors could potentially be mapped to the same point in the unspecified latent subspace. This step directly drives the encoder to retain only information about the unspecified factors in the z variables.

The variational loss (3.1) enables sampling of the unspecified latent variables and aids the

generation of novel images. However, the encoder does not necessarily learn a unique mapping from the image space to the unspecified latent space. In other words, training samples with similar unspecified factors are likely to get mapped to significantly different unspecified latent variables. This observation motivates our *pairwise* reverse cycle loss (3.2), which penalizes the encoder if the unspecified latent embeddings z_1'' and z_2'' have a large pairwise distance, but not if they are mapped farther away from the originally sampled point z_i. This modification is in contrast with the typical usage of cycle consistency in previous works.

$$\min_{Enc} \mathcal{L}_{reverse} = \mathbb{E}_{x_1,x_2\sim p(x),\ z_i\sim\mathcal{N}(0,I)}[|| f_{z_i}(Dec(z, f_s(x_1)))f_z(Dec(z_i, f_s(x_2))) ||_1] \tag{3.2}$$

Chapter 4

Experiments and Results

4.0.1 Quality of generated images

This particular section contains the set of experiments which demonstrate and compare our model's (Jha et al. [3]) capability to generate images for the following experiments :

4.0.1.1 Style-Transfer Experiments

Here we compare our model (Jha et al. [3]) with Mathieu et al. 's [1] and Szabó et al.'s model [2] on the grounds their ability to transfer style for MNIST dataset and further demonstrate our model's ability to generate compelling style-transfer results for MultiPIE dataset with cross-dataset evaluation on UTK-face dataset [27] and CelebA dataset. The results are referred and organized below.

1. MNIST: Figure 4.1
2. MultiPIE: The experiments for MultiPIE dataset with images having cropped for the faces and resizing the given images to 128x128 were carried out with specified factors as:
   (a) Identity: Figure 4.8.
   The model was trained on 20 unique identities with `3000:300` images per identity defining the `train:test` split.
   (b) Pose: Figure 4.12.
   The model was trained on 16 unique poses of 20 distinct identities with `200:20` images per identity per pose defining the `train:test` split.
   (c) Expression : Figure 4.13.
   The model was trained on 6 unique expressions on MultiPIE 51K dataset with `80:20` `train:test` split. Cross-dataset evaluation is performed on the UTK-face dataset.

Figure 4.1: Image generation results on MNIST by swapping z and s variables. The top row and the first column are randomly selected from the test set. The remaining grid is generated by taking z from the digit in first column and s from the digit in first row. This keeps the unspecified factors constant in rows and the specified factors constant in columns.

4.0.1.2 Linear interpolation in the manifolds

We show the variation captured in the two latent manifolds of our models by linear interpolation for MultiPIE dataset : Figure 4.9. A smooth transition in both the identity and the style space stresses on the fact that a linear interpolation in the learnt latent manifolds is a geodesic in the facial image space.

4.0.1.3 Random sampling / Query

We show the reconstructions obtained on passing the latent manifolds with fixed specified domain(extracted from the test images) and a randomly sampling the unspecified domain through the decoder for MultiPIE dataset : Figure 4.10. This particular experiment validates the model's capability to be used as a generative data-augmentation setup - to create more labelled data.

4.0.1.4 Few-shot learning results

This is similar to the "Both specified and unspecified factors from unseen identities" experiment performed in the style-transfer sub-section. Here we study how completely unseen images from the same dataset are captured by our model for both MNIST : Figure 4.14, MultiPIE : Figure 4.15, and CelebA (which is a completely unseen dataset for the model) : 4.16 datasets.

4.0.2 Quality of disentangled representations:

This particular section contains the set of experiments demonstrate and compare our model's (Jha et al.'s [3]) quality of disentangled latent subspaces.

4.0.2.1 Classifier Accuracy

We trained a two layer MLP classifier separately on both the specified and unspecified latent subspaces generated by each model. Since the specified features of variation are associated with the available class labels in a dataset, the classifier accuracy is a good metric to account for the amount of specified-feature specific information in the two latent subspaces. If the specified factors of variation were completely disentangled, the classification accuracy in the specified latent space should be as high as possible, while that in the unspecified latent space should be close to chance. We also compare and check for the **robustness to dimensionality change** of all the three models for MNIST dataset and CMU-MultiPIE[1](Table 4.1).

4.0.2.2 Visualizing the t-SNE plots

Visualizing the t-SNE plots of both the unspecified (without reparameterization) and the specified latent subspaces. Similar to the above conjecture, the unspecified latent space t-SNE plots should be close to a randomly scattered plot to imply effective disentangling of the specified feature, on the other hand should be perfectly clustered for the specified latent subspace. The results for MultiPIE dataset are placed in Figure 4.11.

[1]Since the training was done on a subset of the CMU-MultiPIE dataset (with 20 unique identities with `3000:300` images, for which all the possible variations were captured, per identity defining the `train:test` split) thus, this performance is not meant to be taken as the state-of-the-art performance.

Classification Accuracies						
Iterations	**z dim**	**s dim**	**z train acc.**	**z test acc.**	**s train acc.**	**s test acc.**
MultiPIE - Identity						
5,000	64	256	11.6252	12.0244	99.9983	99.9320
5,000	128	512	11.8289	12.3131	99.9749	99.7792
MultiPIE - Pose						
5,000	576	64	10.9658	10.7167	99.7729	98.7432
MultiPIE - Expression						
2,000	64	64	34.4543	30.2176	99.9916	83.8727
MNIST						
100,000	16	16	18.0171	18.0588	99.8330	98.2872
100,000	32	32	17.0890	16.9571	99.9165	98.6177
100,000	64	64	17.1440	17.3577	99.9365	98.5777
100,000	128	128	19.1022	18.8902	99.9565	98.5176
100,000	256	256	18.8100	18.7500	99.9699	98.3874
100,000	512	512	20.5528	20.2223	99.9198	98.2672
MNIST - LeCun						
100,000	16	16	69.4911	65.5749	99.6861	98.5777
100,000	32	32	63.5583	58.9743	99.9332	98.6278
100,000	64	64	71.0386	67.2876	99.9732	98.7279
100,000	128	128	62.2946	59.0845	99.3940	98.3974
100,000	256	256	60.7872	58.8241	99.9332	98.6478
100,000	512	512	59.9976	58.2732	99.8864	98.3173
MNIST - Challenges						
100,000	16	16	61.7671	48.8181	99.4207	97.9667
100,000	32	32	85.3148	71.8149	99.7696	97.9767
100,000	64	64	99.3422	92.4879	99.9348	98.5977
100,000	128	128	99.1135	96.7447	99.5259	98.4575
100,000	256	256	99.9949	98.0268	99.9666	98.1169
100,000	512	512	99.9966	98.3373	99.9816	98.2672
MNIST - Wasserstien Distance						
100,000	16	16	16.7935	16.3361	99.8063	98.3273

Table 4.1: Quantitative results for the MultiPIE and MNIST robustness test experiments. Classification accuracies on the z and s latent spaces are a good indicator of the amount of specified factor information present in them. Since we are aiming for disentangled representations for unspecified and specified factors of variation, *lower is better* for the z latent space and *higher is better* the s latent space.

4.0.3 Some experiments for valuable insights

With the aim to improve the image reconstruction quality of cycle consistent `VAE` we study the effect of the following experiments : modifying the proposed loss function, and replacing vanilla `VAE` with `PixelVAE` [25] in the model. The results for these experiments are validated in the following fashion:

- **Quantitative Results :** We trained a two layer MLP classifier separately on both the specified and unspecified latent subspaces generated by each model. Since the specified features of variation are associated with the available class labels in a dataset, the classifier accuracy is a good metric to account for the amount of specified-feature specific information in the two latent subspaces. If the specified factors of variation were completely disentangled, the classification accuracy in the specified latent space should be as high as possible, while that in the unspecified latent space should be close to chance.

  This is to make sure that while improving the image reconstruction quality, the change in loss function does not worsen the quality of disentangled representations - our primary motive.

- **Style-transfer Results :** The rendered images displaying the transfer of specified features and unspecified features. Where class information or specified features are same across the columns and style information or unspecified features are same across rows are also reported.
  This is to improve the model's capability to be used as a generative setup for data-augmentation.

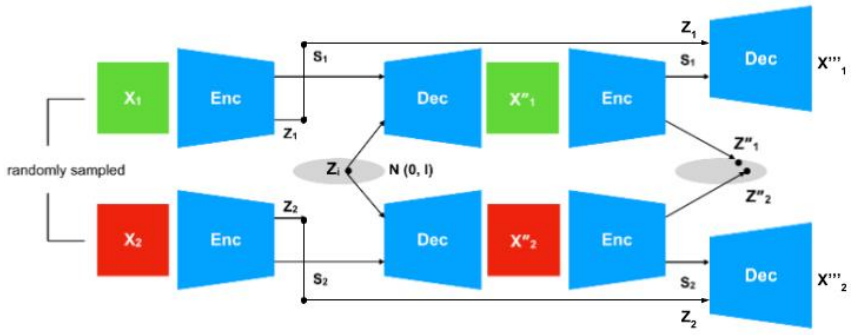


Figure 4.2: Incorporation of reconstruction loss term in the reverse cycle loss.

4.0.3.1 DR-GAN Implementation

With the motive to understand it's working better we debugged the DR-GAN implementation given on github [22] and incorporated the GAN-hacks [9] for the convergence of the model which was trained on CMU MultiPIE dataset and tested on Celebrity Frontal-Profile images. However, the model shows compelling face frontalization results (Figure 4.3), it doesn't disentangle the latent space per se. The decoder is trained to act as a sieve in order to extract the necessary class (here, identity) information to reconstruct the frontal pose image, however, the class/style factors can't be retrieved separately using this particular approach.

Figure 4.3: DR-GAN results. The top row contains the input profile pose images and the bottom row contains the output frontal pose images.

4.0.3.2 Image reconstruction quality improvement

4.0.3.2.1 Loss Function modification :

We propose modified loss functions with the introduction of: (1) adversarial loss and (2) reconstruction loss in addition to existing cycle-consistent loss function for better image reconstruction. We also tried minimizing the (3) wasserstein distance instead of the L1 norm in the reverse cycle.

In the proposed model, the decoder was being trained in the forward cycle to regenerate the input image. While, the encoder was being trained to disentangle the input to specified and unspecified latent factors, in the reverse cycle.

4.0.3.2.1.1 Reconstruction Loss : Owing to the fact that the decoder is trained only in the forward cycle, it might not have been trained sufficiently enough to reconstruct sharper images, so, we thought of introducing a reconstruction loss term in the reverse cycle too. Since, in the reverse cycle, we practically discard the unspecified-latent factors disentangled by the encoder in the first step; we can re-use those by feeding them into the decoder in the last step along with the style information (generated at that step) to generate an image and minimize the reconstruction loss between $X_1 \& X_1'''$ and X_2 & X_2''' [as described in Figure 4.2].

- Quantitative Results : Table 4.2
- Renderings of transferred features : Figure 4.5

4.0.3.2.1.2 Adversarial Loss : With the recent success of adversarial loss functions in generating sharp images, we also tried to improve image reconstruction quality by incorporating an adversarial loss term in the reverse cycle loss. Wherein, (1) **both the decoder and encoder** and (2) **only the decoder** (since, the encoder should focus solely on disentangling the input into specified and unspecified latent factors.); were trained adversarially to make sure that both $X_1 \& X_1''$ and X_2 & X_2'' belong to the same class label by the introduction of a discriminator (labelled Adv (same class)) in the model [Figure 4.4].

- Quantitative Results : Table 4.2
- Renderings of transferred features : Figure 4.6

4.0.3.2.1.3 Minimizing the Wasserstein Distance : Replacing the L1 loss between the two data samples in the reverse cycle with wasserstein distance.

- Quantitative Results : Table 4.2
- Renderings of transferred features : Figure 4.7

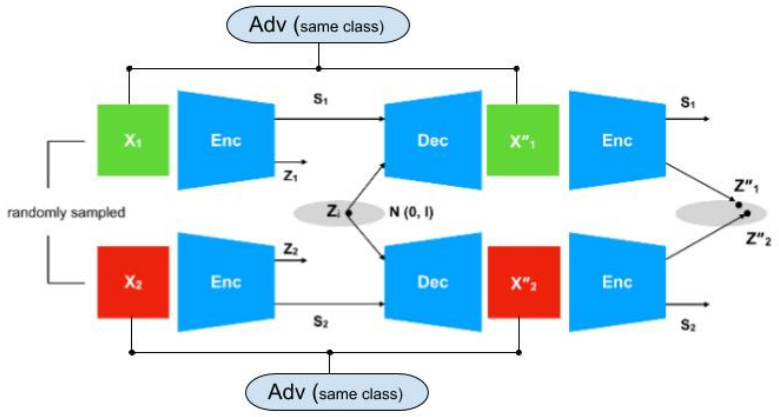


Figure 4.4: Incorporation of adversarial loss term in the reverse cycle loss.

4.0.3.2.2 Replacing vanilla `VAE` with `PixelVAE` :

We tried replacing the vanilla `VAE` with `PixelVAE` [25], a `VAE` implementing a tractable likelihood function (unlike the approximate estimated lower bound likelihood function for vanilla `VAE`s) using an auto-regressive decoder, in the proposed model. On implementing `PixelVAE` in pytorch, we found no significant improvements in the image reconstruction quality when compared to that of a vanilla `VAE` for MNIST dataset; the comparision was done for both random sampling

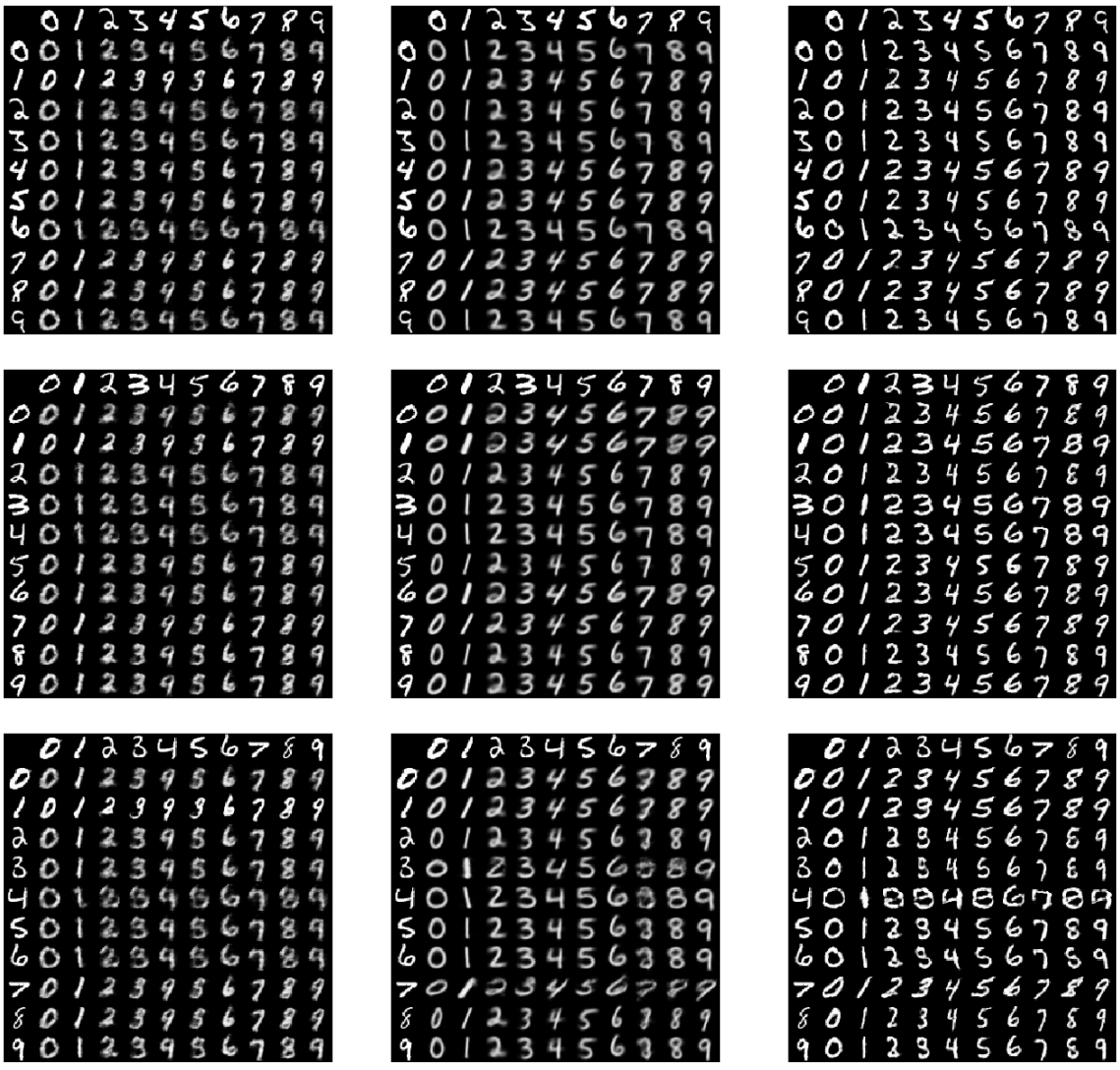
Architecture	z dim	s dim	z train acc.	z test acc.	s train acc.	s test acc.
Incorporating Adversarial Loss (MNIST)						
Vanilla	16	16	17.72	17.56	99.72	98.35
Only Decoder	16	16	18.356	18.038	98.35	97.005
Both	16	16	23.36	23.277	99.058	97.906
Incorporating Reconstruction Loss (MNIST)						
With	16	16	14.09	14.25	99.89	98.377
Without	16	16	17.72	17.56	99.72	98.35
Wasserstien Distance in Loss (MNIST)						
With	16	16	16.7935	16.3361	99.8063	98.3273
Without	16	16	17.72	17.56	99.72	98.35

Table 4.2: Experiments for better insights - Quantitative results for all the experiments performed. Here, z is the unspecified domain class representation and s is the specified domain class representation. Since we are aiming for disentangled representations for unspecified and specified factors of variation, lower accuracy is better for the z latent space and a higher accuracy is better for the s latent space.

[Figure 4.17] and image reconstructions [Figure 4.18] (owing to it's auto-regressive nature, and hence substantial image generation time, `PixelVAE` reconstructions were not compared to vanilla `VAE` on MultiPIE).

(a) Without Reconstruction Loss (b) With Reconstruction Loss

Figure 4.5: Renderings of transferred features, for Jha et al's model with / without the reconstruction loss term in the reverse cycle loss. Where class subspace is same along the columns and style subspace is same along the rows. Note: row containing the digit '7' shows the best comparison amongst the two approaches.

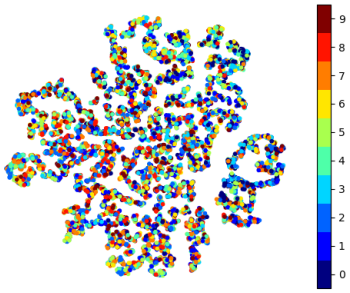


(a) Only Decoder (b) Vanilla (c) Both encoder and decoder

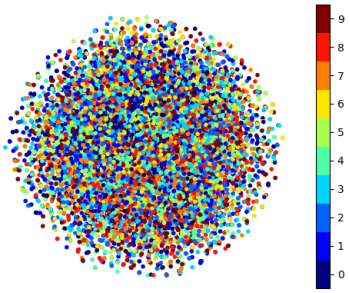
Figure 4.6: Renderings of transferred features, for models with adversarial loss incorporated in the reverse cycle loss function; with columns comprising of renderings for the models (a) where in only the decoder is trained adversarially, (b) vanilla cycle consistent VAE and (c) where in both encoder and decoder were trained adversarially.

(a) With wasserstien distance

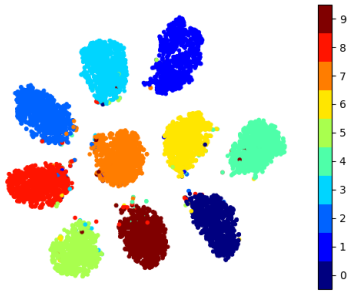
(b) Vanilla or L1 norm



(c) Variation t-SNE without reparam



(d) Variation t-SNE with reparam



(e) Class t-SNE

Figure 4.7: Minimizing the wasserstien distance instead of the L1 norm between the unseen class samples in the reverse cycle.

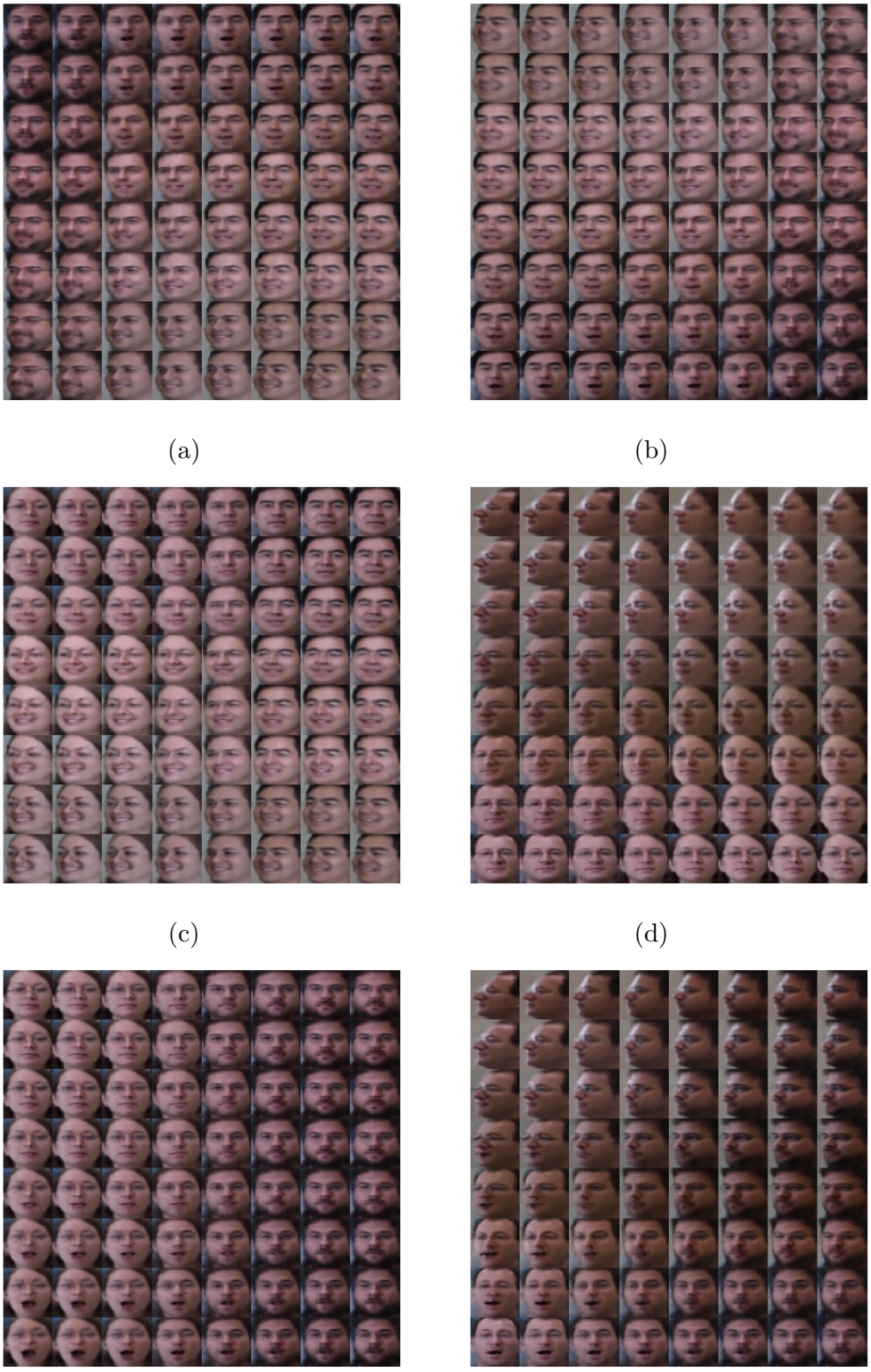
(a) Unseen images for seen identities

(b) Unspecified domain from unseen identities

(c) Specified domain from unseen identities

(d) Both specified and unspecified domain from unseen identities

Figure 4.8: Renderings of transferred features (for unspecified domain dimentionality = 128 and specified domain dimentionality = 512), for both models on MultiPIE dataset with specified domain being identity for (a) Unseen images for seen identities, (b) Unspecified domain from unseen identities, (c) Specified domain from unseen identities and (d) Both specified and unspecified domain from unseen identities. Here the specified component or identity is same along the columns and unspecified domain is same along the rows.



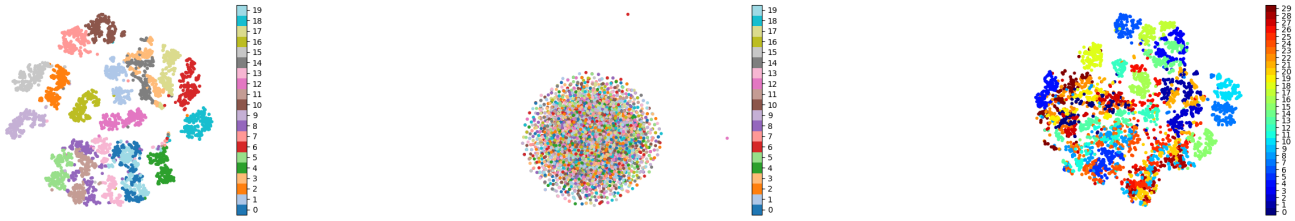
(a) (b) (c) (d) (e) (f)

Figure 4.9: Linear interpolation results for our model (for unspecified domain dimentionality = 128 and specified domain dimentionality = 512) in the z and s latent spaces. The images in the top-left and the bottom-right corner are taken from the test set. Like Fig. 4.8, z variable is constant in the rows, while s is constant in the columns.

(a) (b)

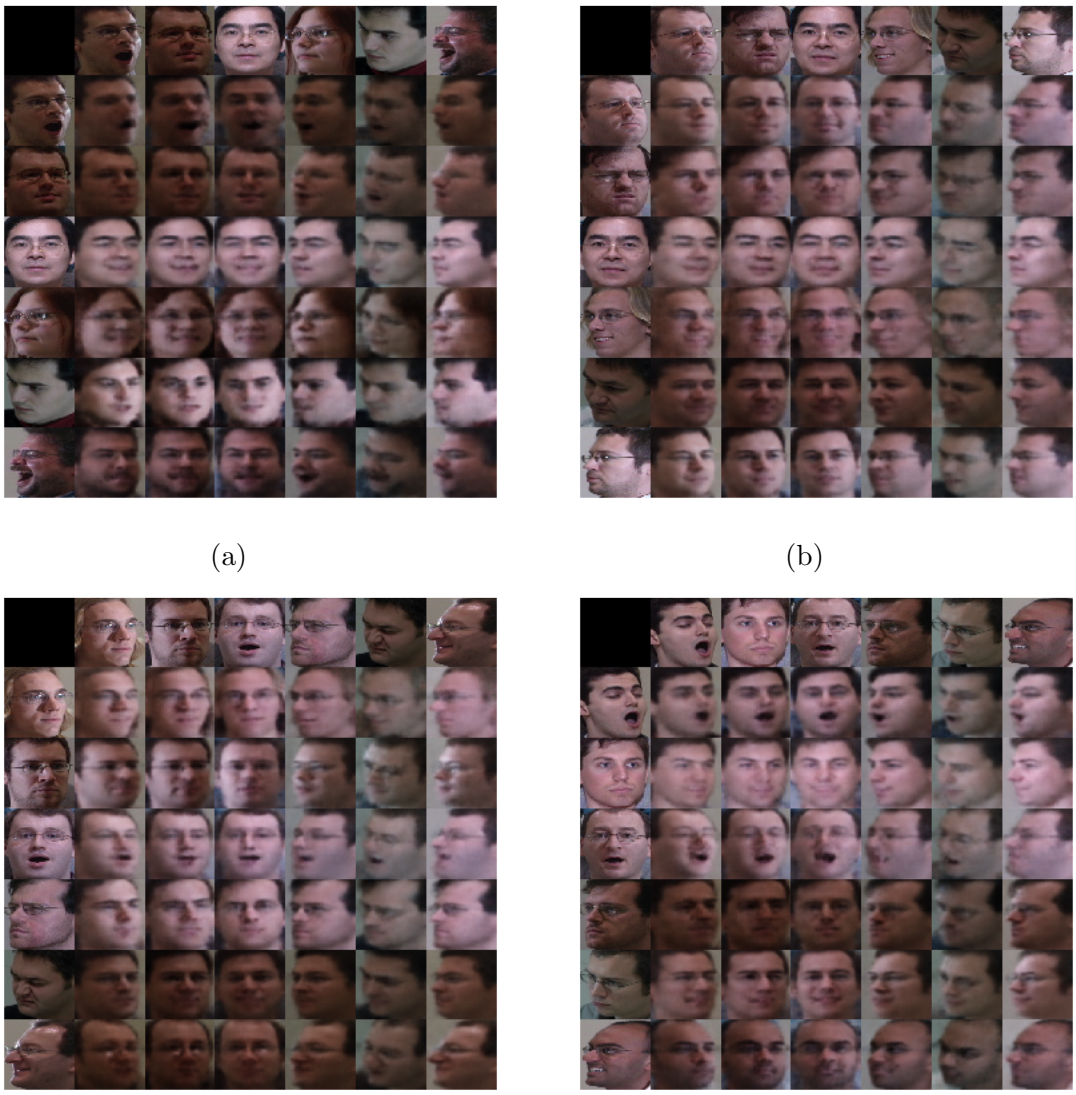
(c) (d)

Figure 4.10: Reconstructions obtained on passing the latent manifolds with fixed specified domain(extracted from the test images) and a randomly sampling the unspecified domain through the decoder of our model (for unspecified domain dimentionality = 128 and specified domain dimentionality = 512). Here the specified component or identity is same along the columns and unspecified domain is same along the rows.



(a) Specified subspace of test images (b) Unspecified subspace of test images (C) Specified subspace of all classes.

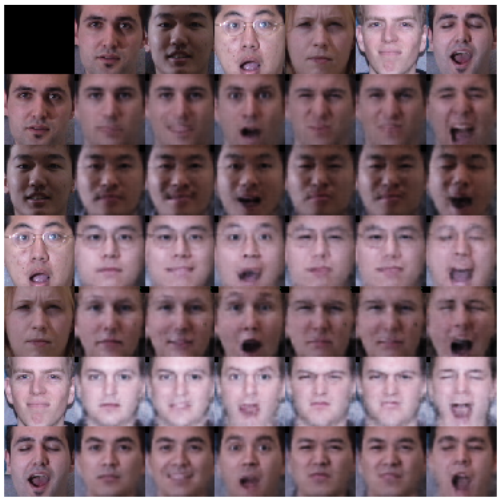
Figure 4.11: t-SNE plot visualization for MultiPIE dataset with specified factor as identity for unspecified domain dimentionality = 128 and specified domain dimentionality = 512. (a) Plot of specified subspace for test images i.e. unseen images of seen classes (here identity), (b) Plot of unspecified subspace for test images i.e. unseen images of seen classes (here identity) and (c) Plot of specified subspace for all the classes - seen as well as unseen; here 0 to 19 (blue to light green) are the seen classes and 20 - 29 (yellow to red) are unseen classes. It is observed that the specified component of the unseen classes falls in between the already seen classes in a rather unstructured fashion - hence, the reconstructions for the same in Figure 4.8 (d) are close to those of the seen identities, since the decoder tends to recreate an average identity for the unseen data point based on it's neighbouring seen classes.



(a) (b)

(c) (d)

Figure 4.12: Renderings of transferred features (for unspecified domain dimentionality = 576 and specified domain dimentionality = 64), for both models on MultiPIE dataset with specified domain being pose for test images. Here the specified component or identity is same along the columns and unspecified domain is same along the rows.



(a) Test images Expression Transfer

(b) Cross-Dataset Expression Transfer

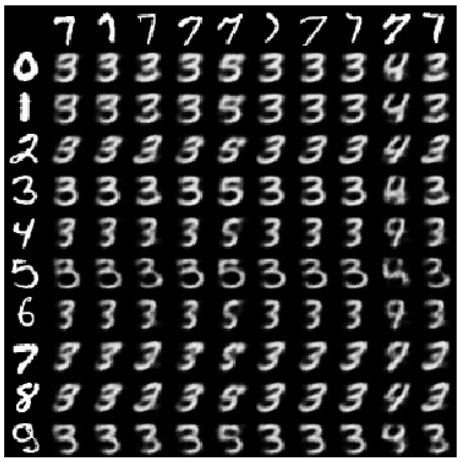
Figure 4.13: Renderings of transferred features (for unspecified domain dimentionality = 576 and specified domain dimentionality = 64), for both models on MultiPIE dataset with specified domain being expression for (a) Test images (unseen images) and (b) Expression transfer from UTK-face dataset (not at all trained on). Here the specified component or identity is same along the columns and unspecified domain is same along the rows.

(a) Trained with one unseen class image

(b) Trained with one unseen class image

(c) Trained with ten unseen class image

(d) Trained with ten unseen class image



(e) Trained with zero unseen class image

Figure 4.14: Few-Shot experiments with unseen class = 7, 8, 9. Training solely 7.

(a) Trained with zero unseen class image

(b) Trained with zero unseen class image

(c) Trained with one unseen class image

(d) Trained with one unseen class image

(e) Trained with one unseen class image

(f) Trained with one unseen class image

Figure 4.15: Few-Shot experiments with unseen class image being the last image in the first row and the first column.

(a) Trained with zero unseen class image

(b) Trained with zero unseen class image

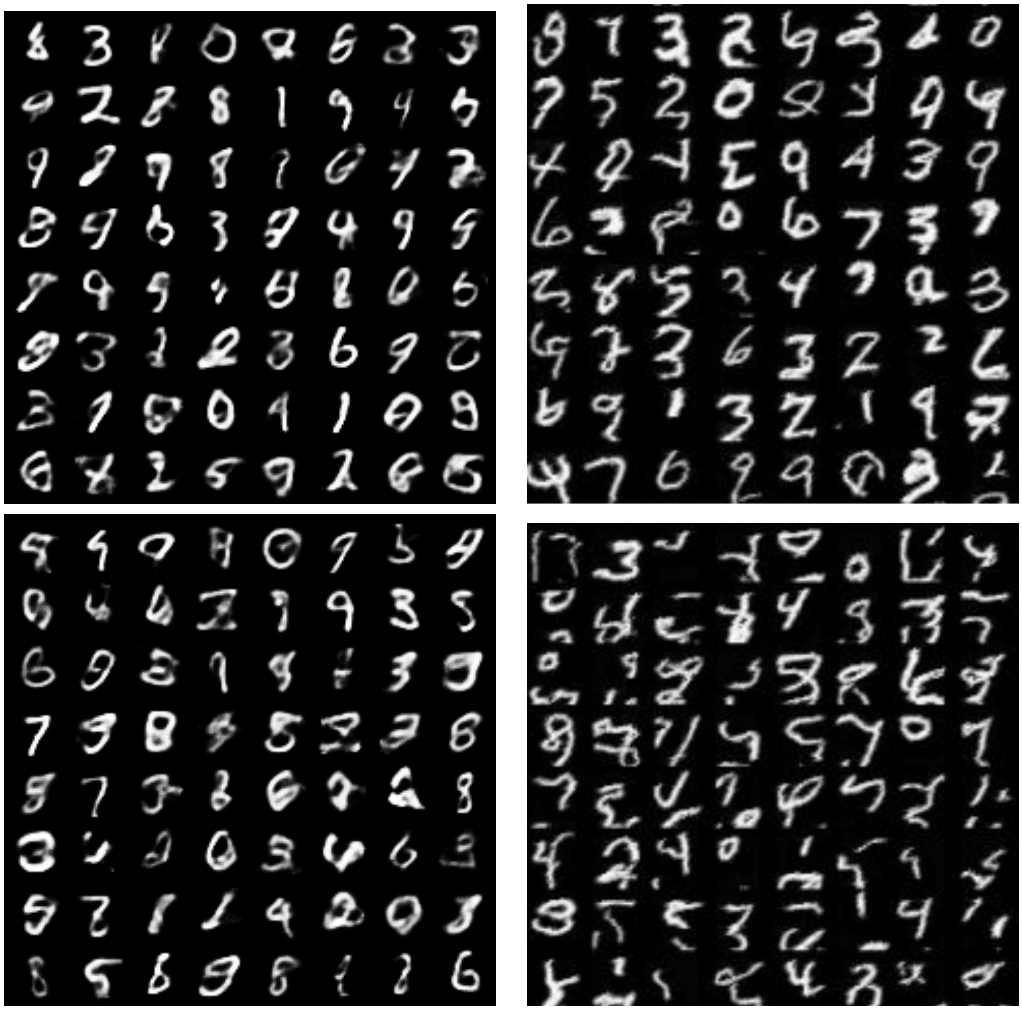
(c) Trained with one unseen class image

(d) Trained with one unseen class image

(e) Trained with ten unseen class image

(f) Trained with ten unseen class image

Figure 4.16: Few-Shot experiments with unseen class image (from CelebA dataset) being the last image in the first row and the first column.



(a) Vanilla VAE (b) PixelVAE

Figure 4.17: Random sampling results comparison for PixelVAE and vanilla VAE.

Figure 4.18: Image reconstruction comparison between PixelVAE and vanilla VAE. First row is the input image, second contains the vanilla VAE reconstructions and third the PixelVAE reconstructions.

Chapter 5

Conclusion

For disentangling, our model (Jha et al. [3]) work disentangles the pose and identity information better than Mathieu et al.'s work - the sate-of-the-art adversarial approach. It is evident from the quantitative results (table 4.1), and the t-SNE plots; however, due to the variational nature of the model proposed the renderings of transferred features produces blurry outputs.

DR-GAN [21] generates compelling results in the target pose domain, but doesn't disentangle the specified domain information, the decoder is trained to act as a sieve in order to extract the necessary identity information to reconstruct the frontal pose image, however, the class/style factors can't be retrieved separately using this particular approach.

Image reconstruction quality improvement by incorporating loss function modification gave us a better insight as to how the proposed model works the best; with the encoder's sole purpose being disentangling the input image to unspecified and specified latent factors and decoder being the only one responsible for composing and reconstructing these disentangled components. Results for adversarial loss incorporation suggest that training both the encoder and decoder performed better than just training the decoder or vanilla model in terms of image reconstruction quality, but, it degraded the disentanglement quality compared to that of the vanilla cycle consistent disentangling model (by 5% increase of the classifier accuracy in the unspecified domain). On the other hand, incorporation of reconstruction loss improved the quality of disentanglement and the quality of image reconstruction (which is most evident when comparing the '7' digit row in figure 4.5).

On implementing PixelVAE, we found no significant improvements in the image reconstruction quality when compared to that of a vanilla VAE for MNIST dataset.

Chapter 6

Future Work

The following could be done to improve the work even further:

- Improving the image regeneration quality, with no adverse effect on the classification accuracies of both unspecified and specified domain, i.e. with no adverse effect to the structure of the disentangled representations obtained, , with or without retaining the non-adversarial nature of the proposed model.
- Improving the results for the few-shot learning problem and optimizing it to be a zero-shot learning model.

Bibliography

[1] Mathieu M., Zhao J.J., Sprechmann P., Ramesh A., LeCun Y. (2016) Disentangling Factors of Variation in Deep Representation using Adversarial Training. NIPS. 50415049

[2] Szab A., Hu Q., Portenier T., Zwicker M., Favaro, P. (2017) Challenges in Disentangling Independent Factors of Variation. arXiv preprint arXiv:1711.02245

[3] Ananya Harsh Jha, Saket Anand et al. (2018) Disentangling Factors of Variation with Cycle-Consistent Variational Auto-Encoders. Accepted in ECCV 2018

[4] Bourlard, Herve & Kamp, Y. (1988). Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. Biological cybernetics. 59. 291-4. 10.1007/BF00332918.

[5] G. E. Hinton and R. R. Salakhutdinov (2006) Reducing the Dimensionality of Data with Neural Networks.Science.

[6] Kingma D.P., Welling M. (2014) Auto-Encoding Variational Bayes. International Conference in Learning Representations. ICLR2014

[7] Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A.C., Bengio Y. (2014) Generative Adversarial Nets.NIPS.

[8] Radford A., Metz L., Chintala S. (2016) Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.International Conference in Learning Representations. ICLR2016

[9] Chintala S. : GAN Hacks

[10] Makhzani, A. Shlens J., Jaitly N., Goodfellow I.(2016) Adversarial autoencoders. International Conference on Learning Representations.

[11] Mehdi Mirza, Simon Osindero (2014) Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784

[12] Jost Tobias Springenberg (ICLR 2016) Unsupervised and semi-supervised learning with categorical generative adversarial networks. arXiv preprint arXiv:1511.06390

[13] Ghahramani, Z.: Factorial Learning and the EM Algorithm. Proceedings of the 7th International Conference on Neural Information Processing Systems. NIPS94, Cambridge, MA, USA, MIT Press (1994) 617624

[14] Tenenbaum, J.B., Freeman, W.T.: Separating Style and Content with Bilinear Models. Neural Computation 12(6) (2000) 12471283

[15] Desjardins, G., Courville, A.C., Bengio, Y.: Disentangling Factors of Variation via Generative Entangling. arXiv preprint arXiv:1210.5474 (2012)

[16] Reed, S.E., Sohn, K., Zhang, Y., Lee, H.: Learning to Disentangle Factors of Variation with Manifold Interaction. In: ICML. Volume 32 of JMLR Workshop and Conference Proceedings., JMLR.org (2014) 14311439

[17] Tang, Y., Salakhutdinov, R., Hinton, G.E.: Deep Lambertian Networks. In: ICML, icml.cc / Omnipress (2012)

[18] Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.B.: Deep Convolutional Inverse Graphics Network. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2. NIPS15, Cambridge, MA, USA, MIT Press (2015) 25392547

[19] Edwards, H., Storkey, A.J.: Censoring Representations with an Adversary. In: International Conference in Learning Representations. ICLR2016

[20] Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.S.: The Variational Fair Autoencoder. In: International Conference in Learning Representations. ICLR2016

[21] Tran, L., Yin, X., Liu, X.: Disentangled Representation Learning GAN for Pose- Invariant Face Recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)

[22] kayamin : DRGAN implementation

[23] Nataniel Ruiz, James M. Rehg : Dockerface: an Easy to Install and Use Faster R-CNN Face Detector in a Docker Container. arXiv preprint arXiv:1708.04370 (2017).

[24] Peiyun Hu, Deva Ramanan (2016) Finding Tiny Faces. arXiv preprint arXiv:1612.04402.

[25] Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, Aaron Courville (2017) PixelVAE: A Latent Variable Model for Natural Images. ICLR 2017

[26] Cian Eastwood, Christopher K. I. Williams (2018) A Framework for the Quantitative Evaluation of Disentangled Representations. ICLR 2018

[27] UTK-Face dataset