Learning by Generating Mental Imagery from Limited Labeled Dataset

Esube T. Bekle[1], **Wallace Lawson**[2]
[1]NRC Postdoctoral Fellow
[2]Navy Center for Applied Research in Artificial Intelligence
US Naval Research Laboratory
Washington, DC 20375
esube.bekele.ctr@nrl.navy.mil

Abstract

Most real-world scenarios such as surveillance and life-long learning on mobile robotic platforms are characterized by a scarcity of labeled training datasets for deep learning. Training deeper networks on limited labeled datasets is non-trivial. Weaker supervision has the potential to alleviate this problem in part. We propose deep convolutional generative adversarial networks (DCGAN) that learn to produce a "canonical image" from input images using an internal representation of the expected distributions of the input data. This canonical image is what the DCGAN "imagines" that the input image category might look like under ideal conditions, i.e. iconic representation. A DCGAN learns this association by training an encoder to capture salient features from the original image and a decoder to convert salient features into its associated canonical image representation. Our new approach, which we refer to as a Mental Image DCGAN (MIDCGAN), learns features that are useful for classification, and that this in turn has the benefit of helping single or few shot recognition from limited labeled datasets. We demonstrate our approach on object instance recognition and handwritten digit recognition tasks.

1 Introduction

Despite the early success of deep convolutional neural networks in computer vision, training deeper networks with limited labeled datasets is a challenge. In such situations, leveraging strategies that are similar to how humans learn would be beneficial. Consider the way that children interact with objects when they are very young (6; 15): during their interaction, children look at objects from different perspectives. Eventually, they build up a preference for certain viewpoints after examining objects for a long period of time. They use that preferred viewpoint of the object as their preferred 'canonical image' atlas or internal representation of the object. When a similar object is encountered from a different viewpoint, they try to map it to the canonical preferred viewpoint they formed. This process is mostly with little to no supervision. In this paper, we consider the question of what would happen if we were to train a deep convolutional generative adversarial network in the same manner (see Fig. 1). For this, we provide the canonical image (i.e., an ideal representation for a category), to map the image to its canonical representation. The form of this supervision is a high-level and weak. Rather than using labels, instead we use the image of the object at different viewpoints. The Mental Image DCGAN (MIDCGAN) is trained to associate each of these samples in a specific input distribution back to the canonical image. This association is learned using a GAN architecture (see Fig. 2) with a generator composed of an encoder and decoder. MIDCGAN trains the encoder to learn salient bottleneck features for each class while the decoder learns to generate a canonical image from bottleneck features. We will show that MIDCGAN bottleneck features are better suited for

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

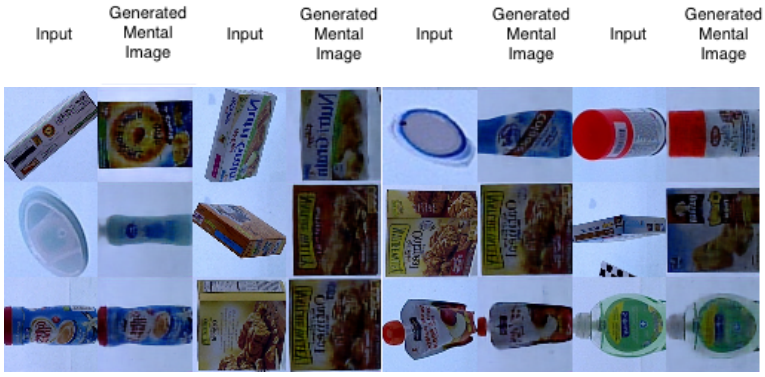

Figure 1: MIDCGAN input (first column and every other column) and generated canonical image (second column and every other column). The images are selected from the BigBird database (12).

learning than those features that are generated without the benefit of using a canonical image (or regular DCGAN).

Stated formally, a typical learning task seeks to learn the data distribution, $p(x)$ mapping to a class or category label y or $p(y|x)$. The diversity of samples in the training data are limiting in the way the learner represents the category class internally. The MIDCGAN approach on the other hand provides a canonical image $\tilde{x}$ as target to be learned and stored as representation of a category expected target distribution. Hence, the diversity of the input samples will no longer be a problem (as the ideal canonical or preferred representation is supplied in the form of the canonical image) and the network can be trained with limited training samples with decreased performance degradation. The learner maps the input data distribution, $p(x)$, to this canonical representation of the category, $\tilde{x}$, i.e the conditional $p(\tilde{x}|x)$. During this mapping process, the MIDCGAN creates an internal bottleneck feature vector that is better representative of the input distribution mapping to the canonical image. In summary, the canonical image forces the network to focus on the iconic parts and features of the object.

Generative adversarial networks (GAN) (2) and their variants such as improved GAN (11), categorical GAN (13) have been shown to learn features for classification. MIDCGAN is inspired partly by the work of image-to-image mapping (5) and image editing (14) and inpainting using GANs (9). DCGANs have been employed to learn expressive features for classification in unsupervised (10) and semi-supervised manner (11; 13).

We demonstrate the effectiveness of mental image DCGANs (MIDCGAN) on three different problems. First, we demonstrate this for handwritten digits as proof of concept. In this case, we assume that a helpful tutor has provided an ideal representation of the digit, so the canonical image is a stencil. Next, we demonstrate the performance of MIDCGAN on instance based object recognition, in which, the helpful tutor provides the system with an iconic view of the object. Finally, we evaluate performance on a dataset that was not used to learn salient features. In all cases, we show that MIDCGAN substantially outperforms its DCGAN equivalent.

2 Methodology

DCGAN learns generative models using the joint adversarial training of a generator network and discriminator network. The generator maps a noise vector z to generated image using $\hat{x} = G_{\theta^G}(z)$ that increasingly represents the underlying target distribution $p(x)$ during training. The discriminator predicts whether a sample is from the real target data distribution, $p(x)$, or the generated data distribution, $p(\hat{x})$, and is represented by $D_{\theta^D}(x, \hat{x})$ (11). For the sake of simplicity, we refer to the generator as $G(z)$ and the discriminator as $D(x)$. We train the generator network using both l_2 loss and the adversarial discriminative loss so that it learns features useful for classification. Unlike DCGAN, the generator network is not generating samples from a random noise vector, z, rather they are derived from the encoder of an autoencoder (9). This enables MIDCGAN to encode important features from the input distribution, $p(x)$, into a bottleneck feature vector denoted as $\bar{z}$.

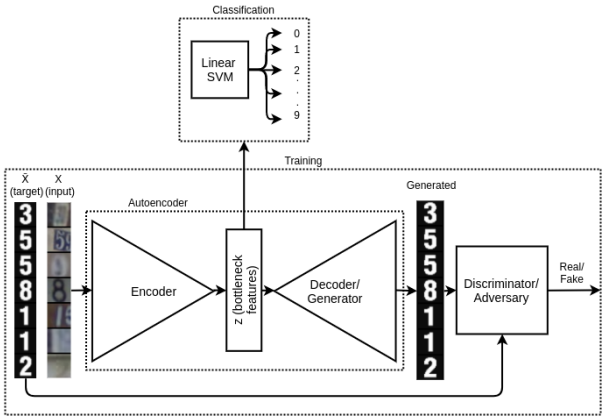

Figure 2: The MIDCGAN architecture built with either feedforward convolutional or ResNet blocks.

2.1 Architecture

The MIDCGAN architecture is based on DCGAN with the generator replaced by a full autoencoder (i.e., encoder-decoder). Given an input selected from a distribution $p(x)$, the encoder network produces a representation of the object in the form of a bottleneck feature vector, $\bar{z}_n$, of length n. The decoder network then takes the output of the encoder, $\bar{z}_n$, and produces the a sample from the generated distribution, $p(\hat{\tilde{x}})$. Here, $\tilde{x}$ is the canonical image of x.

Fig. 2 shows an overview of the MIDCGAN architecture. The convolutional blocks in all the three components of MIDCGAN (encoder, decoder and discriminator) take two forms: either simple convolutional blocks that contain a sequence of convolution-batch normalization-activation (we call this simple network) or n residual blocks. We use $n = 5$ double convolutional BN-activation-convolution residual units (3) in each of the convolution blocks. The discriminator is of two types: an AlexNet style discriminator (similar to most DCGANs) and a discriminator that has the same architecture as the encoder. We refer to these as the AlexNet and Separate discriminators.

2.2 The Joint Autoencoder and Adversarial Loss

MIDCGAN is trained using joint adversarial and generator l_2 losses. In a regular autoencoder, the input and the target are the same and since there is no guidance for the encoder to select a specific mode of the multimodal underlying data distribution, $p(x)$. Therefore, the autoencoder tends to average modes resulting in a blurry average image (9). By adding the canonical image as a target in the l_2 loss component, it forces the network to produce a specific mode of the target distribution, $p(\tilde{x})$, instead (2; 9). Therefore, MIDCGAN generates images that are sharper and closer to the target earlier in the training than an equivalent regular DCGAN while at the same time learning useful classification features via its mapping of the input sample to the canonical image.

2.2.1 Autoencoder Reconstruction Loss

Here we used a normalized l_2 loss similar to (9) without the masking. The l_2 loss is given by the squared difference between the target canonical image, $\tilde{x}$, and the generated canonical image, $G(z)$. Substituting encoded features from input data distribution, $p(x)$, for z using the encoder, $E(x)$ results in Eq. 1

$$L_{l_2}(x, \tilde{x}) = \mathop{\mathbb{E}}_{x \in p(x), \tilde{x} \in p(\tilde{x})} ||\tilde{x} - G(E(x))||_2^2 \tag{1}$$

2.2.2 Adversarial Loss

According to (2), the regular adversarial two-player minmax game between the discriminator, $D(x)$, and the generator, $G(z)$ is given by Eq. 2.

Table 1: MNIST test error (%) with n labeled examples for different architectures at nBn=256. Note: SS is simple feedforward CNN encoder/decoder/discriminator and separate discriminator and RS is residual CNN encoder/decoder/discriminator and separate discriminator

Method/ Arch	Number of Examples					All
	10	20	50	100	200	
DCGAN Alexnet	53.39 $\pm$ 4.22	47.1 $\pm$ 3.63	34.44 $\pm$ 2.13	24.55 $\pm$ 1.56	19.54 $\pm$ 1.1	8.17
Springenber et. al. (13)	-	-	-	1.39 $\pm$ 0.28	-	**0.48**
Makhzani et. al. (8)	-	-	1.90 $\pm$ 0.1	-	-	0.85
Salimans et. al. (11)	-	16.77 $\pm$ 4.52	2.21 $\pm$ 1.36	**0.93 $\pm$ 0.07**	**0.9 $\pm$ 0.04**	-
MIDCGAN SS	**1.51 $\pm$ 0.27**	**1.22 $\pm$ 0.08**	**1.13 $\pm$ 0.09**	1.07 $\pm$ 0.08	0.99 $\pm$ 0.11	0.82
MIDCGAN RS	4.58 $\pm$ 2.65	1.72 $\pm$ 0.69	1.48 $\pm$ 0.43	1.17 $\pm$ 0.13	1.03 $\pm$ 0.07	0.68

$$\underset{G}{min}\underset{D}{max} \underset{x\in p(x)}{\mathbb{E}}[log(D(x))] + \underset{z\in p(z)}{\mathbb{E}}[log(1 - D(G(z)))] \tag{2}$$

In Eq. 3, $p(x)$ represents the actual data distribution of the dataset input to the encoder while $p(\tilde{x})$ represents the target canonical canonical image distribution.

$$L_{adv}(x, \tilde{x}) = \underset{\tilde{x}\in p(\tilde{x})}{\mathbb{E}}[log(D(\tilde{x}))] + \underset{x\in p(x)}{\mathbb{E}}[log(1 - D(G(E(x))))] \tag{3}$$

In practice (2; 9), this loss is implemented by training the discriminator and the encoder-generator pair using alternating SGD. The total joint loss used to train the encoder-decoder generator pair is the weighted sum of the Eq. 1 and Eq. 3 and is given as Eq. 4.

$$L_{total} = \lambda_{l_2} L_{l_2} + \lambda_{adv} L_{adv} \tag{4}$$

The training of MIDCGAN is shown in Algorithm 1. The convergence of MIDCGAN happens often early due to the dual mode selection because of the canonical image and the adversarial loss and hence the discriminator could tell a real target sample quickly.

Algorithm 1 Training MIDCGAN

$\theta^E, \theta^G, \theta^D \leftarrow HeNormal$ (3) ▷ initialize encoder, decoder/generator and discriminator params
repeat
  $X, \tilde{X} \leftarrow$ shuffled mini-batch ▷ X is the input image while $\tilde{X}$ is the canonical image
  $Z \leftarrow Encoder(X)$
  $\hat{\tilde{X}} \leftarrow Decoder(Z)$
  $L_{adv} \leftarrow log(D(\tilde{X})) + log(1 - D(\hat{\tilde{X}}))$ ▷ Compute adversarial loss for real and fake
  $\theta^D \leftarrow \theta^D - \nabla_{\theta D}(L_{adv})$ ▷ Update Discriminator params with the adversarial loss gradients
  $l_2 \leftarrow ||\tilde{X} - \hat{\tilde{X}}||_2^2$
  $L_T \leftarrow \lambda_{l_2} l_2 + \lambda_{adv} L_{adv}$ ▷ total loss as fraction of adversarial and l_2 losses
  $\theta^E \leftarrow \theta^E - \nabla_{\theta E}(L_T)$ ▷ Update Encoder params with the total loss gradients
  $\theta^G \leftarrow \theta^G - \nabla_{\theta G}(L_T)$ ▷ Update Decoder/Generator params with total loss gradients
until number of epochs

3 Results and Discussion

The bottleneck features z represent important aspects of the input distribution, $p(x)$. During test time, we need only the encoder to generate the bottleneck features $\bar{z}$. These features are then given to an l_2 SVM to perform classification. In this section, we evaluate classification accuracy on handwritten digit recognition (MNIST and SVHN) and object instance recognition (BigBIRD and RGBD).

3.1 MNIST with Stencils 'Canonical Images'

In this experiment, the encoder, decoder and the separate discriminator were all ResNet architectures with each convolution block replaced by 5 dual-convolutional residual units. Therefore, the separate discriminator is much deeper and more expressive than the more commonly used AlexNet discriminator. In Table 1, we compare different architectures trained with a bottleneck size of 256 against the

Table 2: SVHN test error (%) with n labeled examples for different architectures at nBn=2048.

Method/ Arch	Number of Examples		
	10	**100**	**1000**
Makhzani et. al. (8)	-	-	17.70± 0.3
Salimans et. al. (11)	-	-	8.11± 1.3
MIDCGAN SA	**17.8± 4.11**	**8.4± 0.68**	**7.15± 0.26**

Table 3: Object recognition accuracy using different databases and approaches.

Dataset	Method/ Arch	Pre-training	Examples per Class				
			50	**25**	**10**	**5**	**1**
BigBIRD	MIDCGAN	Self-supervised	**98.19 ± 0.14**	**95.45 ± 0.39**	**87.4 ± 0.59**	**79.8 ± 0.89**	**52.24 ± 0.64**
BigBIRD	DCGAN	Unsupervised	65.55 ± 0.42	54.7 ± 0.45	43.4 ± 0.54	32.6 ± 0.73	14.1 ± 0.89
RGBD	MIDCGAN	Transfer	**61.7 ± 0.25**	**59.7 ± 0.61**	**55.99 ± 0.54**	**51.4 ± 0.36**	37.6 ± 0.8
RGBD	DCGAN	Transfer	43.33 ± 0.5	40.34 ± 0.51	37.5 ± 0.46	32.1 ± 0.35	20 ± 0.8
RGBD	CNN (4)	Supervised	-	-	-	-	**63.9**

state-of-the-art (13; 8; 11) with a different number of labeled training samples to highlight limited supervision performance.

3.2 SVHN with Stencils 'Canonical Images'

MIDCGAN with simple feedforward blocks and AlexNet style discriminator outperforms all other architectures we experimented with for the SVHN dataset. As shown in Table 2, MIDCGAN produced the best performance among all the methods with a wide range of number of limited training samples.

3.3 Object Instance Recognition

In object instance recognition, rather than identifying a general class of objects (eg., a bottle), we instead identify the specific instance of the object (e.g., a coke bottle). We evaluate this on the BigBIRD (12) and the RGBD (7). The former dataset has 125 object instances, whereas the later has over 300 objects instances. The RGBD dataset has a depth component, but in our experiments we only used RGB data. We evaluate this in two different ways. In the first, we permitted the MIDCGAN to see the objects in the training set, with the first image chosen arbitrarily as the canonical image. In the second, MIDCGAN did not see any training image from RGB-D, but rather to use features that it had already learned in the BigBIRD dataset. We present both results in Table 3. For the first case, we have results with varying numbers of examples per class. With only a single instance of each class (i.e., *single shot recognition*), MIDCGAN recognizes objects 52.2% of the time, compared to 14% of the time for DCGAN. MIDCGAN still significantly outperforms the DCGAN in the second case of transfer learning.

4 Conclusion

In our experiments, we demonstrated the ability to use canonical images to improve recognition with deeper networks when a limited amount of training is available. Although MIDCGAN was demonstrated on representative viewpoints, the selection of the mental target image could be arbitrary. In cases when MIDCGAN was not sure about the object, the generated image did not resemble the actual representative image, or important details about the representative image were omitted. In essence, this could potentially provide another way to determine the confidence in the prediction of the object class. Several improvements could be made to MIDCGAN in the future including autonomous selection of the canonical images (including a set of mixture of canonical images with different views of the same object), and incorporating limited labels into the MIDCGAN network loss function. With such approach and an autonomous mobile robot, for instance, a wide range of objects could be learned with minimal supervision by incorporating only the available labels as another to the loss function and just by generative and adversarial components of the loss for those which do not have labels.

Acknowledgements

Wallace Lawson was supported by the Office of Naval Research, Esube Bekele was supported by the National Research Council.

References

[1] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bülthoff, and C. Wallraven. Active object recognition on a humanoid robot. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2021–2028. IEEE, 2012.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[4] D. Held, S. Savarese, and S. Thrun. Deep lerning for single-view instance recognition. *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.

[6] K. H. James, S. S. Jones, L. B. Smith, and S. N. Swain. Young children's self-generated object views and object recognition. *Journal of Cognition and Development*, 15(3):393–401, 2014.

[7] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[8] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[10] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2226–2234, 2016.

[12] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel. Bigbird: A large-scale 3d database of object instances. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 509–516. IEEE, 2014.

[13] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[14] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *European Conference on Computer Vision*, pages 318–335. Springer, 2016.

[15] C. Yu, L. B. Smith, H. Shen, A. F. Pereira, and T. Smith. Active information selection: Visual attention through the hands. *IEEE Transactions on Autonomous Mental Development*, 1(2):141–151, 2009.