DeGAN: Data-Enriching GAN for Retrieving Representative Samples from a Trained Classifier

Sravanti Addepalli, Gaurav Kumar Nayak, Anirban Chakraborty, R. Venkatesh Babu
Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India
{sravantia, gauravnayak, anirban, venky}@iisc.ac.in

Abstract

In this era of digital information explosion, an abundance of data from numerous modalities is being generated as well as archived everyday. However, most problems associated with training Deep Neural Networks still revolve around lack of data that is rich enough for a given task. Data is required not only for training an initial model, but also for future learning tasks such as Model Compression and Incremental Learning. A diverse dataset may be used for training an initial model, but it may not be feasible to store it throughout the product life cycle due to data privacy issues or memory constraints. We propose to bridge the gap between the abundance of available data and lack of relevant data, for the future learning tasks of a given trained network. We use the available data, that may be an imbalanced subset of the original training dataset, or a related domain dataset, to retrieve representative samples from a trained classifier, using a novel *Data-enriching GAN* (DeGAN) framework. We demonstrate that data from a related domain can be leveraged to achieve state-of-the-art performance for the tasks of Data-free Knowledge Distillation and Incremental Learning on benchmark datasets. We further demonstrate that our proposed framework can enrich any data, even from unrelated domains, to make it more useful for the future learning tasks of a given network.

1 Introduction

The performance and generalizability of Deep Neural Networks largely depend on the amount and quality of training data available. Several successful implementations of tasks such as classification, object detection and segmentation leverage very large, class-balanced and diverse datasets. In addition to training an initial network, data is also required for future updates to the model. This makes it important for training data to be available throughout the life cycle of a product. While data collection is a challenge in itself, storing the data for future use could also be a concern due to data confidentiality constraints, privacy issues, or memory costs. Business establishments may also protect their data to gain market advantage. There are several corporates that provide trained models to clients while maintaining the confidentiality of their data. The client may want to compress the model (Hinton, Vinyals, and Dean 2015; Srinivas and Babu 2015) before deployment, expand the functionality of the model (Rebuffi et al. 2017), or refine the model based on domain specific data (Csurka 2017). These tasks typically require access to the initial training data, which may not be available to the client due to its proprietary nature. In addition, storage and future use of personal data could be limited due to privacy restrictions. This could include user details, biometric data or medical history.

Non-availability of data restricts future enhancements to a trained model. This issue has fuelled research in limited-data and data-free learning approaches to specific tasks such as Knowledge Distillation (Lopes, Fenu, and Starner 2017), Incremental Learning (Castro et al. 2018) and improving model robustness (Mopuri, Krishna, and Babu 2018). The key concern with Data-free approaches is that they operate in a severely constrained setting, where they assume non-availability of any additional data. This often leads to the process of reconstructing samples using variants of activation maximization (Erhan et al. 2009), which is computationally expensive. Several iterations of back-propagation are required to generate one batch of representative samples. One of the concerns with few-shot learning approaches is that they require careful selection of exemplars (Castro et al. 2018), which may not be permitted in a privacy restricted setting. We aim to bridge the gap between data-free approaches, which over-constrain the problem setting; few-shot learning approaches, which assume availability of cherry-picked samples; and the traditional learning methods, that assume the entire training dataset to be available; by using related domain data as a proxy to the true data.

While the lack of a rich, diverse dataset that is relevant to a given application is a key challenge, there are ever increasing sources of diverse data available on the web, which can potentially be tapped for the same. Such data however cannot be used directly as they may not belong to the same distribution as the training dataset, may not be diverse enough, and may not be equally represented across all classes. In this paper, we propose a Data-enriching GAN (DeGAN) framework to enrich *any available data* to make it more useful for the future learning tasks of a pre-trained classifier. We term the *any available data* used as *Proxy Data*, or *Proxy Dataset* as it serves as a proxy to the *True dataset*. *Proxy dataset* could comprise of unlabeled test data collected over a limited duration, or open source datasets, or a collection

of images from the web, or synthetic images. In most practical industry applications, if a trained model is being enhanced for future use, it will have access to unlabeled test data. However, this test data may not be diverse enough, and it may contain data only from a few classes. Using such data directly for tasks such as Knowledge Distillation would lead to very poor performance. DeGAN can enrich this data to make it more representative of the training data, and introduce the diversity that is crucial for future learning tasks.

As an example, a real-life use case of cancer screening is considered here, where an initial teacher model is trained using a large corpus of CT-scans of patients across various geographies. Training data is rich in terms of diversity and has class balance. However, the sensitivity and size of this data forbids its storage for future use. The teacher model is deployed for a few years, after which there is a requirement of deploying it on a handheld device with lower memory and compute. The organization may decide to use one month data from a given hospital to train the student net. This data is very much related to the true dataset, however it lacks the richness and diversity. It is also possibly class-imbalanced (containing non-cancerous classes only), leading to a degraded performance of the distilled network. Our proposed DeGAN is not only capable of generating a diverse set of samples, but can also handle the class-imbalance problem by using only one class data to generate representative samples for all classes.

In some applications such as Class-Incremental Learning, the availability of *Proxy Dataset* is not an additional requirement. Here, an initial model is trained on old classes, which is incrementally trained on new class data in future. In a data-free scenario, old class data is assumed to be unavailable. Here, DeGAN can use the new class data as *Proxy Dataset* to retrieve representative samples related to old classes from the pre-trained network.

The organization of this paper is as follows: The subsequent section outlines our contribution in this paper. This is followed by a discussion on the existing literature related to our work. Section 4 gives a detailed description of our proposed approach. Following this, we present our experiments and results in Section 5. We conclude the paper with our analysis of the proposed method in Section 6.

2 Contributions

In this work, we propose a novel approach to retrieve representative samples from a pre-trained classifier using a three-player adversarial framework. We use a *Proxy Dataset*, which is data from a related domain, but may be class-imbalanced, or composed of partially overlapping/ non-overlapping classes, as an aid to retrieve these samples. The three players in our proposed architecture are generator, discriminator and pre-trained classifier. In addition to the adversarial game between generator and discriminator that exists in a conventional GAN (Goodfellow et al. 2014) setup, we introduce an adversarial play between the discriminator and classifier as well. While discriminator tries to bring the distribution of the generated data closer to that of the related domain (*Proxy Dataset*), classifier tries to bring in features specific to the original training data distribution (*True Dataset*). The classifier also ensures class balance in the generated samples. The result of this three-way adversarial training is that the generated samples lie on the image manifold of the *Proxy Dataset*, while they also incorporate features from the *True Dataset*.

We consider the task of Knowledge Distillation to demonstrate that data from a related domain can be leveraged to achieve state-of-the-art performance for the future learning tasks of a pre-trained network. The process of generating samples is agnostic to any future task where they would potentially be used. This allows the generated data to be used for multiple tasks. We demonstrate proof-of-concept for this claim by considering the task of single-step Class-Incremental learning for CIFAR-100 dataset. Our contribution in this work can be summarized as follows:

- We propose a *Data-enriching GAN* (DeGAN) framework to retrieve representative samples from a trained classifier using data from a related domain.
- We demonstrate state-of-the-art performance on the task of Data-Free Knowledge Distillation on CIFAR-10 (Krizhevsky and Hinton 2009) and Fashion MNIST (Xiao, Rasul, and Vollgraf 2017) datasets using data generated by DeGAN.
- We are the first to show results for Data-Free Knowledge Distillation on a dataset of larger size (CIFAR-100). This demonstrates the scalability of our data generation approach, when compared to the existing methods.
- We show that the proposed DeGAN could enrich data even from an unrelated domain, to make it more useful for the future learning tasks of a given network (such as using SVHN dataset for retrieving data from a CIFAR-10 classifier)
- We demonstrate state-of-the-art performance for the task of Data-free single-step Class Incremental Learning as a proof of concept of applicability of DeGAN to multiple Machine Learning tasks.

3 Related Works

There are several works in existing literature, related to extracting representative samples from a trained classifier for various applications. Simonyan, Vedaldi, and Zisserman (2013) retrieve class specific samples from a deep convolutional network by maximizing their class scores, for the purpose of visualization. Similar ideas have been used by Mopuri, Krishna, and Babu (2018) for creating samples that are representative of a class, and using them for the task of crafting adversarial perturbations. These methods are computationally expensive as several iterations of back propagation are required to construct a single sample. Also, the generated samples are usually specific to a given task. For example, they may not be diverse enough to train a neural network and get optimal performance. Our proposed approach is task agnostic and is capable of generating samples that are diverse enough for tasks such as Knowledge Distillation and Class-Incremental Learning.

3.1 Knowledge Distillation

Knowledge distillation is a technique of transferring knowledge from a large capacity network (*Teacher*) to a smaller network (*Student*) without significant impact on accuracy. Early methods (Hinton, Vinyals, and Dean 2015) utilize the entire training data for the task of distillation. Li et al. (2018) use 1% of the training data to train the *Student* model. Kimura et al. (2018) create pseudo samples and augment them with few samples of the training data to train the *Student* network. The pseudo samples are updated by increasing *Student-Teacher* loss, whereas the network parameters are learned to reduce the same loss through an iterative and complicated optimization process. Here, the process of generating representative samples is task specific, as opposed to our proposed approach.

Lopes, Fenu, and Starner (2017) use metadata to perform Knowledge Distillation. The statistics of training data are saved at each layer in the form of activation records, which are utilized to reconstruct the training samples. The work by Nayak et al. (2019) is an attempt towards Data-Free Knowledge Distillation, where access to metadata is also not required. The representative samples named *Data Impressions*, are synthesized using the *Teacher* model. Target vectors at the output softmax layer of the *Teacher* network are sampled from a mixture of Dirichlet distributions with carefully selected parameters to ensure diversity. These samples are used to generate images such that the cross-entropy loss between the sampled vector and output of the network (corresponding to the generated images) is minimized. The images generated using this method are used for the task of Knowledge Distillation. Both these approaches require careful selection of parameters and are computationally expensive as multiple iterations of back-propagation are required to generate a single image.

3.2 Class-Incremental Learning

Incremental learning refers to the paradigm of learning continually from a stream of data. In Class-Incremental Learning (Rebuffi et al. 2017), a network is initially trained on a few classes, and is incrementally updated over time to learn new classes. During these updates, the deep model suffers from catastrophic forgetting (McCloskey and Cohen 1989) as it forgets the mapping on old class data when only new class data is used to train the model. While a straightforward method to overcome this issue is to simultaneously train the model using old class data and new class data, this trivial solution is not permitted, as it is assumed to be infeasible to store the old class data due to memory constraints, or other issues discussed earlier in Section 1. This led to using methods such as *finetuning*, which would not allow the model to be updated too much. Another baseline method is *fixed representation*, where the parameters related to old classes are frozen and only the new class parameters are learned using the new data. These methods suffer from low accuracy either on old classes or new classes.

In order to avoid this, Rebuffi et al. (2017) choose a fixed number of exemplars from old class data based on a selection strategy called *herding*, and use them along with new class data to train the incremental model. Cross entropy loss is used on the new class data, whereas distillation loss is used to retain performance on the old classes. Castro et al. (2018) improve the incremental model by jointly learning the data representation and classifier in an end to end fashion. Both these approaches use few exemplars from old class data to avoid catastrophic forgetting, which may not be feasible in a privacy constrained setting.

While Shin et al. (2017) do not use old data for each incremental step, they train a generator using the old data. Our approach has broader applicability as we do not assume the availability of such a generator. Li and Hoiem (2017) assume a Data-Free setting, where there is no access to the old data. The new class samples are used to compute cross entropy loss on new classes, and distillation loss on old classes. While this approach can work well if the new class data is well distributed across all the old classes in the initial Classifier, it would not work in a case where the new class data is highly correlated to a small subset of the old classes. In such a case, the proposed DeGAN can be used to generate class-balanced representative samples of old classes using the new class samples as *Proxy Data*. We demonstrate improved performance as compared to their results, which are reported by Rebuffi et al. (2017) for the Class-Incremental learning task on CIFAR-100 dataset.

We describe our proposed approach in detail in the following section.

4 Proposed Approach

In this section, we first present a classical generative framework for retrieving representative samples from a trained classifier and discuss the associated issues. We further propose constraints that can be imposed to address these issues and discuss existing methods of imposing such constraints. We discuss the benefit of our proposal over the other implementations, followed by a detailed discussion on our proposed DeGAN framework.

4.1 Data-Free Generative approach

The central pathway of Fig. 1(a) shows a classical data-free approach to generating samples using a generator and a pre-trained classifier. Inputs sampled from a latent space are utilised by the generator to produce images which are validated by the classifier. Some of the advantages of such a generative approach with respect to the conventional method of generating samples using activation maximization are, computational efficiency, memory efficiency and better diversity in generated data. However, using a generator could potentially lead to the following issues: mode collapse, and generation of noisy images that do not belong to the data distribution. Nikolaidis et al. (2019) improve the diversity of images by training multiple generators and using all of them for the future goal of Knowledge Distillation. However, this process is inefficient and computationally expensive. Fig. 1(a) illustrates the classical data free generative approach with the required additional constraints. In order to improve diversity, the architecture could include a *diversity enforcing network* whose role is to build a one-to-one mapping from the output space of the generator to the input space.

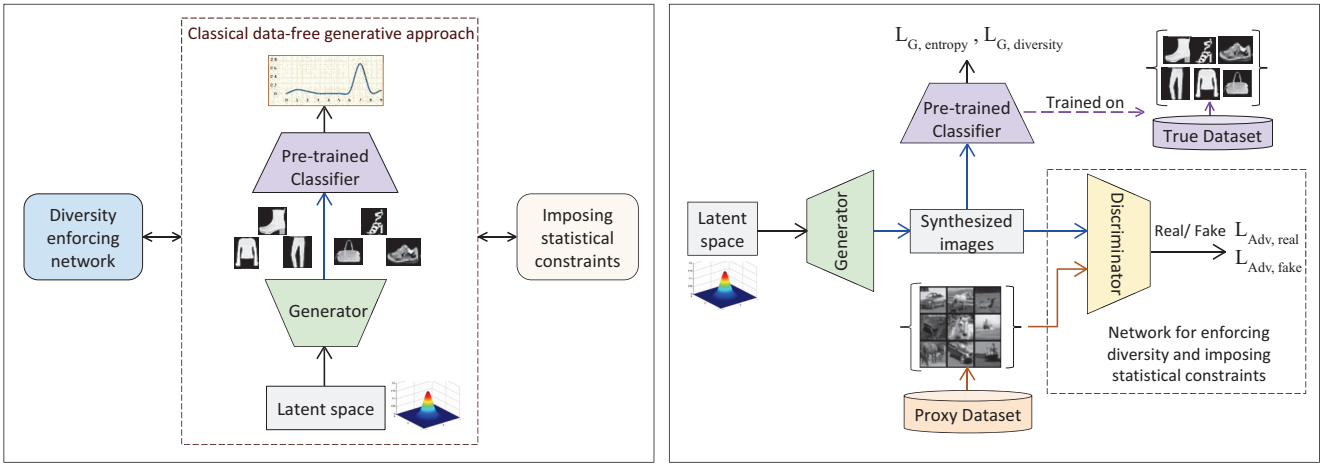

Figure 1: (a) Block diagram showing the classical data-free generative approach with required additional constraints (b) Block diagram of the proposed DeGAN architecture

Since classifier is a many-to-one mapping function, the classical method of retrieving an input based on maximization of output activations can potentially lead to generation of images that are far from the true data distribution. Imposing additional statistical priors on the generated images can move the distribution of same closer to that of the true data. This process is closely related to enforcing visual quality on the generated images and has been well explored in the vision community (Weiss and Freeman 2007; Zeng, Lu, and Borji 2017; Moorthy and Bovik 2011). However, the constraints to be imposed are very specific to the dataset being considered. The process of hand-crafting these constraints can be tedious and assumes a lot of prior knowledge about the original training dataset.

While one can use independent networks or loss functions to impose the constraints discussed above, this would lead to an increase in complexity. We can enforce the same constraints by intelligently utilising a single network (discriminator) in the framework, which motivates the proposed DeGAN, as explained in the following sections.

4.2 Data-Enriching GAN (DeGAN)

Designing novel metrics to impose visual priors individually on each dataset is a challenging task. Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) are known to be useful for imposing priors on images for various tasks in image processing and computer vision. Hence, we introduce a discriminator that serves as the *Imposing statistical constraints* block in Fig. 1(a). Since we assume that the original training data is not available, we use data from a related domain (*Proxy Data*) to train the generator-discriminator pair in an adversarial manner. The rationale behind this is that low level statistics of images remain same or similar for data from the same or related domains. Hence, GAN training ensures that the generated images lie on the manifold of the *Proxy Data*, which is similar to that of the *True Data*. There has been significant progress in the training methods and architectures of GANs to ensure diversity in the generated images (Radford, Metz, and Chintala 2015; Salimans et al. 2016). We leverage the progress in research on this front to address the issue of lack of diversity in the generated samples. Hence, the discriminator also serves the purpose of the *Diversity enforcing network* in Fig. 1(a). We use a Deep Convolutional GAN (DCGAN) (Radford, Metz, and Chintala 2015) for the experiments in this paper as it is very stable to train and scalable. Our proposed method can be used with other GAN architectures as well, and hence can be adapted to various applications.

Using a GAN enables us to generate data that belongs to the distribution of the *Proxy Dataset*. However, we need to ensure that the learned distribution is close to the *True Data* distribution. Therefore, we propose to use a three-player Data-enriching GAN for generating representative samples from a trained classifier. This consists of a Generator, Discriminator and Classifier coupled together as shown in Fig. 1(b). Generator takes a multidimensional random vector as input, with each dimension sampled independently from a standard normal distribution. It generates an image which goes as input to the discriminator and classifier. Here, weights of the generator and discriminator are trainable, while weights of the classifier are frozen. The discriminator ensures that distribution of the generated data is close to that of the *Proxy Dataset*. The role of the classifier is to ensure that the generated data incorporates features that the classifier expects in the input images. Classifier also ensures that the distribution of generated images is balanced across all the classes.

Loss Formulation: We consider $p_z(\boldsymbol{z})$ as the distribution of the latent space (input space of the Generator), $\boldsymbol{z}$ to be a random vector sampled from $p_z(\boldsymbol{z})$, $p_{data}(\boldsymbol{x})$ as the distribution of *True Data*, N as the number of images per batch, K as the number of classes in the *True Dataset*, λ_e and λ_d as positive constants which can be tuned to adjust the weightage of Entropy Loss and Diversity Loss respectively. We

denote the Generator as G, Classifier as C, and Discriminator as D here. We consider $\boldsymbol{y}$ as the Classifier output corresponding to the Generator input $\boldsymbol{z}$. The expectation over Classifier outputs across a batch of samples sampled from the latent space ($p_z(\boldsymbol{z})$) is denoted by $\boldsymbol{w}$.

$$\boldsymbol{y} = C(G(\boldsymbol{z})), \quad \boldsymbol{w} = E_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[C(G(\boldsymbol{z}))] \tag{1}$$

The losses used to train the proposed DeGAN are presented below:

- Adversarial loss (Goodfellow et al. 2014) ($L_{Adv,real}$ and $L_{Adv,fake}$), to ensure that the distribution of the generated images is close to that of the *Proxy Dataset*,

$$L_{Adv,real} = E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[log\ D(\boldsymbol{x})] \tag{2}$$

$$L_{Adv,fake} = E_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[log(1 - D(G(\boldsymbol{z})))] \tag{3}$$

- Entropy loss ($L_{G,entropy}$) (minimization of entropy of individual samples) at the output of the classifier, to ensure that each generated sample belongs to one of the classifier's classes with high confidence,

$$L_{G,entropy} = E_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[-\sum_{k=0}^{K} y_k\ log(y_k)] \tag{4}$$

- Diversity loss ($L_{G,diversity}$) at the output of the classifier, to ensure that the entropy of the expected output of the classifier across a batch is high. When the individual outputs of the classifier are peaky, this loss ensures that the distribution of samples in a batch is uniform across classes. This prevents the generated samples from being biased towards any particular class.

$$L_{G,diversity} = -\sum_{k=0}^{K} w_k\ log(w_k) \tag{5}$$

The equations below describe the Discriminator loss (L_D) and the Generator Loss (L_G). The Generator loss is composed of two additional losses imposed by the Classifier.

$$L_D = L_{Adv,real} + L_{Adv,fake} \tag{6}$$

$$L_G = L_{Adv,fake} + \lambda_e L_{G,entropy} - \lambda_d L_{G,diversity} \tag{7}$$

We alternately maximize L_D (freezing generator parameters) and minimize L_G (freezing discriminator parameters) in a manner similar to the standard GAN training (Goodfellow et al. 2014). A combination of the above losses ensures that the generated images are similar to the related domain, incorporate features from the *True Data* distribution and are equally distributed across all classes of the *True Dataset*.

In cases where the domain of the reference dataset is close to that of the *True Dataset*, the value of λ_e can be low (or set to 0), as the images will already contain some of the features that the classifier expects. However, in cases where the reference dataset is not closely related to the *True Dataset*, we need to give this loss a higher weightage. Class distribution and confidence of the generated images provide cues for tuning these hyperparameters.

Table 1: Accuracy (in %) of Student network using Knowledge Distillation: Comparison with the state of the art ($^{\#}$90 non-overlapping classes from CIFAR-100 are used)

True Dataset	**CIFAR-10**	**F-MNIST**	**CIFAR-100**
Proxy Dataset	CIFAR-100$^{\#}$	CIFAR-10	CIFAR-10
Teacher accuracy	83.02	90.72	79.05
Using *True Data*	81.78	88.98	69.65
Kimura et al.	-	72.5	-
ZSKD	69.56	79.62	-
Proxy Dataset	74.58	77.81	46.32
DCGAN	66.24	79.67	39.77
(Ours) **DeGAN**	80.55	83.79	65.25

In the following section, we demonstrate that data retrieved using our proposed Data-enriching GAN can achieve state-of-the-art performance on the tasks of Knowledge Distillation and Class-Incremental Learning for some of the benchmark datasets. We also demonstrate scalability of our approach to CIFAR-100, which has not been done in any of the existing works.

5 Experiments

In this section, we discuss the experimental setup for an empirical evaluation of our proposed DeGAN framework. We use the benchmark datasets, CIFAR-10 (Krizhevsky and Hinton 2009), Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017) and CIFAR-100 to demonstrate state-of-the-art results for the task of Data-Free Knowledge Distillation. We demonstrate the applicability of our approach to Class-Incremental Learning using CIFAR-100 dataset. We use PyTorch framework for all our implementations.

5.1 Knowledge Distillation Using DeGAN

An overview of the experimental flow is presented below:

- We first train a *Teacher* model (or Classifier) on the *True Dataset*. A train-validation split of 80-20 is considered for this purpose. An early-stopping condition based on validation accuracy is set as the convergence criteria. The weights of the classifier are considered frozen for all future experiments.
- We use this trained classifier in the proposed DeGAN framework to construct representative samples of the *True Dataset*. Samples from the *Proxy Dataset* are used for training the DeGAN. For a given *True Dataset*, we consider experiments with multiple *Proxy Datasets*. We use the implementation of DCGAN from Singh (2019) as reference to implement the DeGAN. The learning rate for training the GAN is set to 0.0002 and a fixed number of epochs are trained (200 in all cases) to ensure consistency. The two hyper-parameters in this case are λ_e and λ_d.
- The trained generator is used to perform the task of Knowledge Distillation. In every epoch of training, we generate a batch of samples using the generator. These samples are used to train the *Student* using Knowledge Distillation(KD) loss (Hinton, Vinyals, and Dean 2015).

Table 2: Accuracy (in %) of Student net with CIFAR-10 as *True Dataset* (column headings indicate the *Proxy Dataset* used)

Methods	CIFAR-100 select samples			CIFAR-100 samples: 10 random classes per sample					SVHN	Random Noise
	90 classes	40 classes	6 classes	Sample-1	Sample-2	Sample-3	Sample-4	Sample-5		
Proxy Data	74.58	65.78	36.44	42.15	49.24	46.65	49.08	47	45.18	11.63
DCGAN	66.24	66.13	39.44	56.81	67.33	62.1	69.34	68.66	26.5	10.09
DeGAN (Ours)	**80.55**	**76.32**	**59.53**	**66.95**	**74.59**	**72.87**	**76.63**	**71.61**	**55.05**	**23.26**

We have given a weight of one to the distillation component of KD loss, and zero to the Cross-Entropy component. This is done to match conditions with the existing works (Nayak et al. 2019), and also to avoid additional hyper-parameter tuning.

- We train a vanilla DCGAN in every case; and use this to perform KD to the *Student* network. This serves as an ablation to prove the usefulness of the third element (Classifier) in DeGAN. The learning rate and number of training epochs are maintained same across the training of DCGAN and DeGAN. We also consider the baseline of using *Proxy Data* directly for the task of Knowledge Distillation. This serves as a lower bound in each case.

Experiments with CIFAR-10 as *True Dataset* CIFAR-10 (Krizhevsky and Hinton 2009) is a 10-class labelled dataset consisting of colour images of size 32×32. This dataset has 50000 training images and 10000 test images. The images are equally distributed across all classes. We consider the *Teacher* architecture as AlexNet and *Student* architecture as AlexNet-half (network with half the capacity when compared to AlexNet), similar to that used by Nayak et al. (2019). With CIFAR-10 as the *True Dataset*, we consider the following *Proxy Datasets*: CIFAR-100 select classes and SVHN. CIFAR-100 (Krizhevsky and Hinton 2009) consists of 100 labelled classes with images of dimension 32×32. Each class has 500 training images and 100 test images. Hence the number of images per class in CIFAR-100 is one-tenth of that in CIFAR-10. The 100 classes in this dataset can be grouped into 20 categories. In this case, CIFAR-100 is a related dataset, since both CIFAR-10 and CIFAR-100 consist of natural images of the same size. Most of the images in both datasets belong to object classes, with a few exceptions in CIFAR-100. Although we consider a related domain dataset, we claim that the classes in the *Proxy Dataset* can be unrelated to those in the *True Dataset*. In order to demonstrate this, we consider multiple combinations of classes as *Proxy Datasets*: (results in Table-2)

1. Only non-overlapping classes between CIFAR-10 and CIFAR-100 are used. The categories, vehicles1 and vehicles2 from CIFAR-100 are excluded here. (90 classes are used)
2. Categories in CIFAR-100 which are even remotely related to the CIFAR-10 classes have been excluded in this case. Categories used are: flowers, food containers, fruits and vegetables, household electric devices, household furniture, trees, large man-made outdoor things, large natural outdoor scenes (40 classes are used)
3. Only background classes are used in this case. The 6 classes used are: Road, Cloud, Forest, Mountain, Plain and Sea. This case can be regarded as an unrelated dataset, since these are not object classes. So, the discriminator in the DeGAN does not learn the notion of an object from the *Proxy Dataset* in this case.
4. 10 classes are randomly sampled from the 40 handpicked unrelated classes and used as *Proxy Dataset*. This random sampling process is repeated 5 times.

In order to understand the true potential of our proposed approach, we consider the case when related datasets are not available. SVHN (Netzer et al. 2011) is a publicly available colour dataset consisting of street view house numbers cropped to the dimension 32×32. This dataset has 10 classes, with each class representing one digit. This dataset consists of 73257 training images and 26032 test images. Since SVHN consists of digits, the statistics of images in this dataset will be different from that in the CIFAR-10 dataset. Hence this is a dataset from an unrelated domain.

The consolidated results of all the above experiments are presented in Table-2. Our results have been compared with the state-of-the-art results in Table-1. We consider the case of CIFAR-100 with 90 classes as the *Proxy Dataset* for this. The performance of our approach is better than the existing work (Nayak et al. 2019) by 10.99% as shown in Table-1. Although the total number of samples used in the 90-class case is 45000, which is lesser than the size of CIFAR-10 training data set, we are still able to closely match the performance of Knowledge Distillation using actual data samples. As we reduce the number of classes in the *Proxy Dataset*, and as we move to classes that are more unrelated to the *True Dataset*, the Knowledge Distillation (KD) accuracy drops. However, our approach is consistently better compared to the two baselines: directly using *Proxy Data* for KD (denoted by *Proxy Data* in Table-2); and using DCGAN to generate samples for KD.

We consider an ablation of using random noise as the *Proxy Dataset*. While the baseline of using *Proxy Dataset* directly gives an accuracy that is close to that of a random guess(11.63%), DeGAN is able to improve the accuracy significantly to 23.26%. The baseline of using DCGAN also gives the accuracy equivalent to a random guess (10.09%). This experiment demonstrates the importance of enforcing a good prior on the generated images. This also demonstrates that the DeGAN framework can enrich *any available Proxy Data* to make it more useful for a given task.

Experiments with Fashion MNIST as *True Dataset* Fashion MNIST (Xiao, Rasul, and Vollgraf 2017) is a grayscale image dataset consisting of 10 object classes. The dimension of each image is 28×28. The dataset consists of 60000 training samples, and 10000 test samples. We con-

sider the *Teacher* architecture as LeNet (LeCun et al. 1998) and *Student* architecture as LeNet-half (network with half the capacity when compared to LeNet), similar to that used by Nayak et al. (2019). We use Fashion MNIST as the *True Dataset* and consider CIFAR-10 and SVHN as *Proxy Datasets*. Both datasets (CIFAR-10 and SVHN) are converted to grayscale and used for training DeGAN. In this case, neither of these datasets belong to the domain of the *True Dataset*. However, CIFAR-10 contains object classes, which is a property that even Fashion MNIST classes possess. Since SVHN has only numbers, it does not possess object-like features. So, CIFAR-10 is more related to the domain of Fashion MNIST dataset when compared to SVHN. We use *Proxy Dataset* as CIFAR-10 for comparison with the state of the art (Table-1). We demonstrate that we can beat the performance of the existing approaches (Kimura et al. 2018; Nayak et al. 2019) by at least 4.17% for the task of Knowledge Distillation using CIFAR-10 dataset, although it is not related to the Fashion MNIST dataset. The margin with respect to the state-of-the-art approach is not as high as the previous case, as we did not consider a related dataset here.

Experiments with CIFAR-100 as *True Dataset* To demonstrate scalability of the proposed DeGAN framework, we use CIFAR-100 as the *True Dataset*. We consider the *Teacher* architecture as Inception-V3 (Szegedy et al. 2016; Chollet 2017) and *Student* architecture as ResNet18 (He et al. 2016). We consider a related dataset, CIFAR-10 as the *Proxy Dataset* here. The number of training data samples in CIFAR-100 is the same as that of CIFAR-10. However, the number of classes is much lesser in CIFAR-10. Although we consider a related dataset with 2 overlap classes (automobiles and trucks), this case is more challenging when compared to the above experiments since the number of classes in the *True Dataset* is larger, and the number of classes in the *Proxy Dataset* is much lesser. This may lead to a high class imbalance when Vanilla-GAN is used, resulting in a large number of misclassifications in the sparsely populated classes. The diversity loss in DeGAN helps maintain balance across the CIFAR-100 classes. The results are presented in Table-1. We see an improvement of about 19% with respect to the baseline. This is a significant improvement for top 1% accuracy of a 100-class dataset.

5.2 Class-Incremental Learning Using DeGAN

We consider the case of single-step Class-Incremental learning on CIFAR-100 dataset. This is done as proof of concept to support the claim that the data generated using DeGAN can be used to replace the *True Dataset* for various tasks. An initial model is first trained on a random set of 20 classes, which are termed as *old classes*. The goal is to incrementally learn the next set of 20 classes in a data-free setting, where the old class data is assumed to be unavailable. We use ResNet-32 (He et al. 2016) architecture for the initial as well as final models, similar to the baselines we compare with (Rebuffi et al. 2017). We use the results reported by Rebuffi et al. (2017) for comparison. We consider the standard losses that are used in Incremental Learning (Li and Hoiem 2017): Cross-entropy loss to learn the *new classes* and Distillation loss, to avoid catastrophic forgetting on the *old classes*. In addition, we also add a regularization term to account for the relative scaling of *logits* between the old and new classes. We use our proposed DeGAN to extract representative samples of *old classes* using the *new class* data as *Proxy Data*. This generated data is used in the Distillation Loss component to avoid catastrophic forgetting of the *old classes*. We compare our results with the baselines explained in Section 3. We also include a baseline of using the *new class* data directly in the Distillation Loss similar to LwF (Li and Hoiem 2017). The results in Table-3 demonstrate a significant improvement in the accuracy with respect to other Data-Free baselines. After one incremental update, since we have a trained generator, existing methods (Shin et al. 2017) that incrementally learn the generator and classifier sequentially, can be used for further updates. This could not be used for the first incremental update as this approach requires data from *old classes* to train the first generator.

Table 3: Single-step Class-Incremental Learning: Comparison with the state of the art

Methods	Accuracy (in %)
Finetuning	41.6
Fixed Representation	46.8
LwF.MC	62.58
Using *Proxy Data*	65.03
DeGAN (Ours)	**68.65**

6 Conclusion

We have proposed a novel Data-enriching GAN (DeGAN) framework to enrich data from any domain, such that it is more suitable for the future tasks of a given trained classifier. The problem of retrieving representative samples from a trained classifier is of importance in several applications such as Knowledge Distillation, Incremental Learning, Visualization and Crafting of Adversarial Perturbations. We have empirically evaluated our framework on several benchmark datasets to demonstrate that we can achieve state-of-the-art results for the task of Data-Free Knowledge Distillation using data from a related domain. We observe that the samples generated using related domain data can also serve as useful visualizations for the *True Dataset*.

We have further demonstrated that the data generated using DeGAN can replace the training dataset for multiple tasks, and hence is truly representative of the same. We show state-of-the-art results on the task of Class-Incremental learning, where we do not have access to old class data.

Since it is not easy to quantitatively judge how similar the domain of the *Proxy Dataset* is, we also evaluate the impact of using unrelated domain data as the *Proxy Dataset*. We show that even if the *Proxy Data* is unrelated to the *True Data*, our proposed DeGAN can significantly enrich the dataset such that it is more useful than using vanilla-GAN for the future tasks.

Acknowledgements This work was partially supported by the Robert Bosch Center for Cyber-Physical Systems (RBC-CPS), Indian Institute of Science (IISc). We would also like to extend our gratitude to the students and research assistants at Video Analytics Lab, IISc for the insightful discussions on this work.

References

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 233–248.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 1251–1258.

Csurka, G. 2017. Domain adaptation for visual applications: A comprehensive survey. *arXiv preprint arXiv:1702.05374*.

Erhan, D.; Bengio, Y.; Courville, A.; and Vincent, P. 2009. Visualizing higher-layer features of a deep network. *University of Montreal* 1341(3):1.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Kimura, A.; Ghahramani, Z.; Takeuchi, K.; Iwata, T.; and Ueda, N. 2018. Few-shot learning of neural networks from scratch by pseudo example optimization. *arXiv preprint arXiv:1802.03039*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Canadian Institute for Advanced Research.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Li, Z., and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence (PAMI)* 40(12):2935–2947.

Li, T.; Li, J.; Liu, Z.; and Zhang, C. 2018. Knowledge distillation from few samples. *arXiv preprint arXiv:1812.01839*.

Lopes, R. G.; Fenu, S.; and Starner, T. 2017. Data-free knowledge distillation for deep neural networks. In *LLD Workshop at Neural Information Processing Systems (NeurIPS)*.

McCloskey, M., and Cohen, N. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory* 24(C):109–165.

Moorthy, A. K., and Bovik, A. C. 2011. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE transactions on Image Processing* 20(12):3350–3364.

Mopuri, K. R.; Krishna, P.; and Babu, R. V. 2018. Ask, acquire, and attack: Data-free uap generation using class impressions. In *European Conference on Computer Vision (ECCV)*, 20–35. Springer.

Nayak, G. K.; Mopuri, K. R.; Shaj, V.; Babu, R. V.; and Chakraborty, A. 2019. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning (ICML)*, 4743–4751.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Nikolaidis, K.; Kristiansen, S.; Goebel, V.; and Plagemann, T. 2019. Learning from higher-layer feature visualizations. *arXiv preprint arXiv:1903.02313*.

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

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001–2010.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *Advances in neural information processing systems (NeurIPS)*, 2234–2242.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2990–2999.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Singh, C. 2019. Pretrained gans in pytorch for mnist/cifar.

Srinivas, S., and Babu, R. V. 2015. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2818–2826.

Weiss, Y., and Freeman, W. T. 2007. What makes a good model of natural images? In *2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. IEEE.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Zeng, Y.; Lu, H.; and Borji, A. 2017. Statistics of deep generated images. *arXiv preprint arXiv:1708.02688*.