Multi-Class Multi-Instance Count Conditioned Adversarial Image Generation

Amrutha Saseendran[1], Kathrin Skubch[1] and Margret Keuper[2]
[1]Bosch Center for Artificial Intelligence, [2]University of Siegen
Amrutha.Saseendran@de.bosch.com

Abstract

Image generation has rapidly evolved in recent years. Modern architectures for adversarial training allow to generate even high resolution images with remarkable quality. At the same time, more and more effort is dedicated towards controlling the content of generated images. In this paper, we take one further step in this direction and propose a conditional generative adversarial network (GAN) that generates images with a defined number of objects from given classes. This entails two fundamental abilities (1) being able to generate high-quality images given a complex constraint and (2) being able to count object instances per class in a given image. Our proposed model modularly extends the successful StyleGAN2 architecture with a count-based conditioning as well as with a regression subnetwork to count the number of generated objects per class during training. In experiments on three different datasets, we show that the proposed model learns to generate images according to the given multiple-class count condition even in the presence of complex backgrounds. In particular, we propose a new dataset, CityCount, which is derived from the Cityscapes street scenes dataset, to evaluate our approach in a challenging and practically relevant scenario. An implementation is available at https://github.com/boschresearch/MCCGAN.

1. Introduction

Developmental studies show that the human brain is endowed with a natural mechanism for understanding numerical quantities [10, 48]. Even young children have an abstract understanding of numeracy and can generalize the concept of counting from one category to another (*e.g.* from objects to sounds) [48]. While counting object instances is relatively easy for humans, it is challenging for deep learning and computer vision algorithms, especially when objects from multiple classes, e.g. persons and cars, are considered. In this paper, we take a step towards such elementary visual reasoning by addressing the generation of images conditioned on the number of object instances per object class. We are particularly interested in the complex case where objects from *multiple classes* are present in the same image (compare Figure 1). This is a fundamental vision task, which can even be solved by small children [10], but remains an unsolved problem in computer vision. Apart from that, many practical applications can benefit from the capability to generate images respecting numerical constraints. Especially, it aids the generation of additional diverse training data for visual question answering and counting approaches. Further, the generation of technical designs based on the number of different components is of particular interest in the field of topology design, where data-based approaches have recently been explored successfully in applications ranging from molecular design [2] for chemical applications to product design [38] for aesthetics or engineering performance.

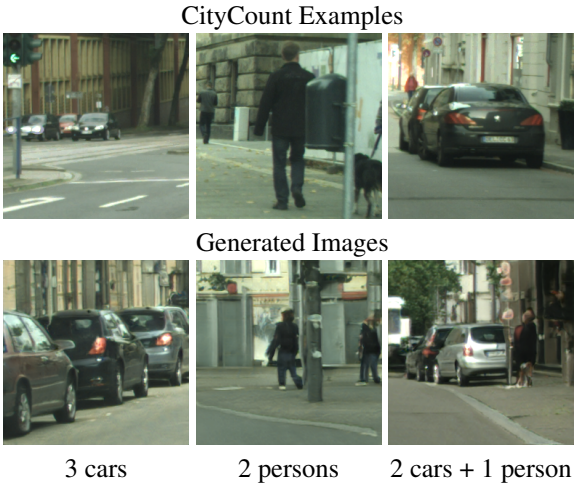


Figure 1: Real and generated CityCount images by our model based on the multiple-class count input.

In this paper, we propose to solve *multiple-class count* (MC^2) conditioned image generation (*i.e.* the generation of images conditioned on the number of objects of different classes, that are visible in the image) as a modular extension to the state-of-the-art network for adversarial image generation, StyleGAN2 [29]. We further argue that object

counting should be considered as a multi-class regression problem. While this approach is simple, it allows the similarity between neighboring numbers to be naturally encoded in the network and to transfer the ability to count from one class to another. This will ideally make our network learn to generalize the concept of counting from one object class to another, meaning that it can for example see images of "two cars and one person" at training time and deduce the appearance of "two persons" at inference time. To the best of our knowledge this is the first attempt to evaluate the potential of GANs to generate images based on the multiple object class count.

We validate the proposed approach in two lines of experiments. First, we evaluate the generative performance of our model on synthetic data generated according to the CLEVR [24] dataset as well as on real data from the SVHN [36] dataset. We further propose a new, challenging real-world dataset, CityCount, which is derived from the well-known street scenes dataset Cityscapes [9]. The CityCount dataset comprises of various crops from Cityscapes images which contain specific numbers of objects from the important classes, *car* and *person*. The dataset includes various challenging scenarios such as diverse and complex backgrounds, object occlusions, varying object scales and scene geometry. Samples from the CityCount dataset and generated samples from our model are shown in Figure 1. In the second line of experiments, we show that the images generated by MC^2-StyleGAN2 can be used to enhance the size and quality of training data for count prediction networks, trained on images from CLEVR and CityCount.

2. Related work

Generative adversarial networks (GANs) - GANs [17] have rapidly evolved to being the most promising trend for the generation of diverse photo-realistic images. Deep convolutional GAN (DCGAN) [39] demonstrated the potential of convolutional neural networks in this context for the first time. A considerable amount of research was devoted to improve the training stability of GANs [18, 26, 33] and to develop more evolved architectures [5, 28, 40]. Conditioning GANs (CGAN) on explicit information was first introduced by Mirza *et al.* [32]. Since then, various approaches have been proposed to improve the controllability of GANs. Many of these require extensive additional information such as class labels and/or natural language descriptions, e.g. image captions for text-to-image or text-to-video generation [4, 20, 32, 41]. Other variants of conditioning GANs include an information-theoretic extension to GANs (InfoGAN) [7], auxiliary classifier GAN (ACGAN) [37], twin auxiliary classifier GAN (TACGAN) [16] and projection based conditioning methods [35]. ACGAN extends the loss function of GAN with an auxiliary classifier to generate images. TACGAN further improves the divergence between real and generated data distribution of ACGAN by an additional network that interacts with both generator and discriminator. In projection based methods [34], the condition is projected to the output of the discriminator by considering the inner product of the conditional variable and the feature vector of images. ContraGANs [25] introduces a conditional contrastive loss to learn the relation between input images. SpatialGAN [20] propose a method for multiple conditioning with bounding box annotations and class labels of objects, and image captions to control the image layout in terms of object identity, size, position and number. In their method, object bounding boxes are provided at test time so the idea of count does not need to be learned. In [11] the authors propose a variational U-Net architecture to condition the image generation on shape or appearance. Various approaches have also been suggested to control the image generation process of GANs in applications such as image-to-image-translation [23, 51] or attribute transfer [19, 31]. Our work is related to ACGAN, with focus on the problem of multiple-class counting using regression.

Based on the high-resolution architecture introduced in [26], StyleGAN [28] employs adaptive instance normalization [22] based feature map re-weighting to facilitate the manipulation of images over multiple latent spaces, encoding different style properties. StyleGAN2 [29] improves over StyleGAN and avoids some characteristic generation artifacts. Recently, a new technique was proposed [27] to achieve state of the art results with StyleGAN2 even when the training data is limited. While these approaches allow implicit conditioning of image contents for example on given styles, they do not enable to steer explicit properties of a generated image such as the number of generated object instances per object class. Our proposed model introduces an extension to StyleGAN2, that facilitates such an explicit conditioning.

Counting approaches - One way to count objects in an image is to first localize and classify them using an object detection network and then count all found instances. While this approach is effective, it also requires additional class labeled bounding box or object prototype information [6, 13, 45]. Adapting these approaches for conditional image generation will require additional information such as pre-defined locations of the objects of interest during training. Other methods rely on recurrent neural network architectures and attention mechanisms [42, 43, 49]. Thus, they can not easily be applied in our problem setting. Density estimation based counting methods [12] show that learning to count can be achieved without prior detection and are more reliable in severe occlusion scenarios. Multiple approaches have been proposed to counting object instances in images, for example in the context of visual question answering [3, 30, 47]. In [1], Agarwal *et al.* suggest to

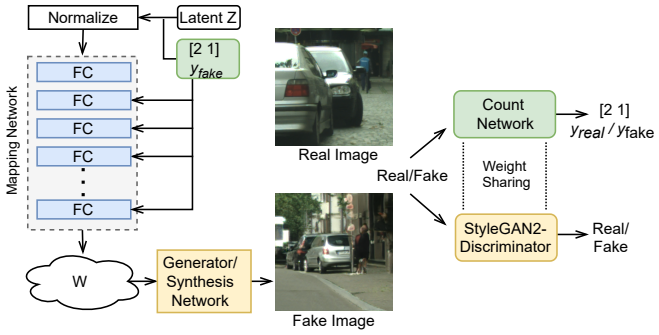


Figure 2: MC^2-StyleGAN2 architecture: The input to the generator is a multiple-class count vector where each index of the vector corresponds to each object class and the value at each index represents the multiplicity of the corresponding object class. In the given CityCount example, the count vector [2,1] corresponds to 2 cars and 1 person respectively.

generate training data for this task by modifying the number of objects using cropping and inpainting. ARIGAN [15] utilizes a conditioned DCGAN to generate images of plants given the number of leaves.

In this paper, we attempt to guide the image generation process solely by conditioning on the number of objects of pre-defined classes in the images, while a reasonable spatial layout is to be inferred from the training data distribution. Instead of addressing single object class counting as seen in [44, 46], where convolutional or recurrent neural networks are used to count digit occurrences, our approach focuses on counting object instances from multiple classes during *generation*. We introduce an extension to the StyleGAN2 architecture by integrating an additional regression network to the discriminator to facilitate image generation based on the number of objects per class. Based on the findings in [8], our network employs dense blocks in the generator architecture to ease the propagation of the count constraint as well as the regression loss of the count network.

3. Multiple class count conditioned image generation

In this section, we introduce the proposed extension to StyleGAN2 for multiple-class count based image generation, MC^2-StyleGAN2.

3.1. MC^2-StyleGAN2

We borrow the architectural specifications of the generator and discriminator from StyleGAN2 and extend the model for our application. The input to the generator is a multiple-class count vector, where each index of the vector corresponds to a different object class and the value at each index represents the number of objects from the corresponding object class. The generative part of our model includes a mapping network to map the combination of latent vector and the count constraint to an intermediate latent vector w and a generator/synthesis network to generate images as shown in Figure 2. To the first layer of the mapping network, we provide a combination of randomly sampled noise and our multiple-class count vector, that specifies which objects and how many of each of them are required in the output image. The count vector is also concatenated to every layer in the mapping network as shown in Figure 2. In the generator network, we introduce dense like skip connections where the output from each block is connected to its succeeding blocks. As shown in Figure 2, the real/generated images are passed through two pathways, (1) an adversarial pathway to classify the input images as real/fake and (2) a count regression pathway, to predict the object class and their multiplicity in the input image. The weight sharing between the two sub-networks regularizes the discriminator and reduces the memory consumption during training.

3.2. Adversarial training with count loss

The generator G, uses both the latent noise distribution $z \sim \mathcal{N}(0,1)$ and a multiple-class count vector $\mathbf{c} = [c_1, c_2, \ldots, c_n]$ that represents n different object classes and their respective multiplicity $c_i, i = , \ldots, n$, to generate fake images $x_{\text{fake}} = G(z, \mathbf{c})$. The discriminator D aims to distinguish between these fake images and real images x_{real}. We denote the data distribution as $x \sim p_{data}(x)$. The additional count sub-network C is trained to predict the per-class object count, y_{fake} for fake images and y_{real} for real images. The adversarial objective of the network is expressed as

$$\begin{aligned}\mathcal{L}_{GAN}(G,D) \quad &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \\ &\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{c})))]. \quad (1)\end{aligned}$$

The multiple-class count loss $\mathcal{L}_{MC^2}$ is defined as the euclidean distance between the predicted count $y_{\text{real}} = C(x_{\text{real}})$ and true count $\mathbf{c}$ of the real images, and the distance between the predicted count $y_{\text{fake}} = C(x_{\text{fake}})$ and the value of the count condition for the generated images.

$$\mathcal{L}_{MC^2}(C) = ||C(x) - \mathbf{c}||_2. \quad (2)$$

The count loss thus enforces the generator to generate images with the desired number of object instances.

Hence, the total loss of the network is a combination of adversarial loss to match the distribution of real images with fake images and a count loss to enforce the network to generate images based on the specified input count. The overall objective function of our method is,

$$\mathcal{L}_{MC^2-StyleGAN2}(G,D) = \mathcal{L}_{GAN}(G,D) + \lambda\mathcal{L}_{MC^2}(C), \quad (3)$$

Method	CLEVR-3		SVHN-2		CityCount	
	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)
SNGAN	0.61	43.68	0.72	47.34	0.55	55.85
ContraGAN	0.68	27.44	0.78	21.12	0.59	49.62
CStyleGAN2	0.65	31.95	0.80	19.42	0.61	13.89
Ours	**0.92**	**8.94**	**0.93**	**10.90**	**0.78**	**8.33**

Table 1: Quantitative analysis across datasets. *For CityCount we used StyleGAN2 with adaptive discriminator augmentation. [27]

where λ steers the importance of the count objective.

Implementation Details The models are trained with images of size 64×64 for SVHN, 128×128 for CLEVR images and 256×256 for CityCount images. All hyperparameters used are provided in the Appendix.

4. Experimental analysis

In the following, we evaluate our model in three different settings. We quantitatively evaluate (1) the ability of the model to predict the multiple-class count in terms of Average count accuracy (Acc) and (2) the quality of the images generated based on the learned count in terms of the Fréchet Inception Distance (FID). The quantitative results of our method (MC^2-StyleGAN2) compared to the state of the art conditional GANs such as SNGAN [34], ContraGAN [25] and Conditonal StyleGAN2 [29] are given in Table 1.

4.1. CLEVR

The objective of the experiment is to analyze the ability of the model to generate complex 3D objects and layouts. The well-known CLEVR dataset comprises images of different 3D shapes, cylinders, cubes and spheres of varying colours. For our experiments, we generate a total of 2000 images for each count combination based on the implementation of CLEVR dataset [24]. We consider two variants of CLEVR images, (1) CLEVR-2 with two shapes, cylinder and sphere, and at most six instances of each shape per image and (2) CLEVR-3 with three shapes, cylinder, sphere and cube, and at most three instances of each shape per image. For our first line of experiments, we consider a simple setting, where we restrict shapes of the same class to be of the same color (red cylinders, green spheres and blue cubes). The generated images shown in Figure 3a show that the proposed model learns to generate images based on the learned object count. For further evaluation, we extend the experimental setting and consider CLEVR shapes with varying colors. As shown in Figure 3b and 3c, the model captures the correlation of the count information even in a more complex setting, where the shape colors do not provide additional information. It can also be observed that the model learned to place objects spatially in reasonable locations although no object bounding box annotations are provided.

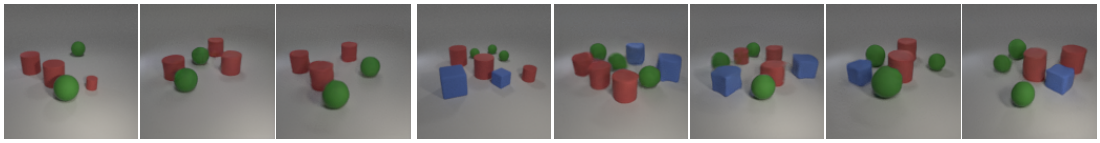
Additionally, for count prediction analysis, we consider the performance of the count sub-network in the model. We observed an average count accuracy of 96% for CLEVR-2 and 92% for CLEVR-3 (a more detailed analysis of the count prediction on CLEVR-2 and CLEVR-3 is provided in the Appendix). For CLEVR-3, the observed count prediction accuracy is comparatively lower than for CLEVR-2, potentially for two reasons, (1) the image distribution is highly complex due to the high number of objects in the image (maximum of nine objects per image) and (2) objects in the images are often overlapping significantly.

Interpolation and Extrapolation We further examine the ability of the model to interpolate between count combinations and to extrapolate to unseen count combinations from one object class to another. For interpolation experiments, we train our model on a subset of CLEVR-2 images, that does not contain images with four spheres and a subset of CLEVR-3 without images of two cylinders, while at test time we evaluate the regression network on exactly such images. The observed count accuracy values for unseen count during testing are 0.94 and 0.91 for CLEVR-2 and CLEVR-3 respectively. This show the potential of the model to transfer the learned count four from cylinders to spheres on CLEVR-2 and the learned count two from spheres and cubes to the cylinder class for CLEVR-3 images. For extrapolation experiments, we train the network with CLEVR-2 images (upto 3 spheres) and plot the success rate in terms of count accuracy and FID to generate 4, 5 and 6 spheres at test time in Figure 4. Here the baseline model is trained with images of spheres and cylinders till count 6. The observed extrapolation performance is comparable to the baseline method. This further confirms that the network is not merely memorizing the count number.

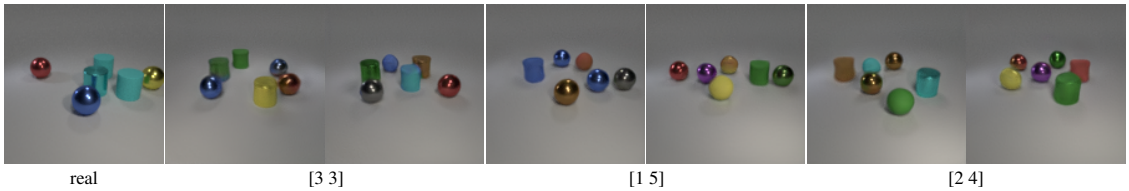
4.2. SVHN

In this section, we consider real world images from noisy training data on the street view house numbers (SVHN) dataset [36]. The dataset includes house numbers cropped from street view images. For our experiments, we considered the original images resized to 64×64 pixels and a total of 1500 samples for each count combination. We restrict ourselves to SVHN images with at most two instances of each digit class (SVHN-2), because images with three or more digits are too scarce for training. The count label is a vector of 10 entries prescribing the multiplicity of each digit in the image. The generated images are shown in Figure 3d.

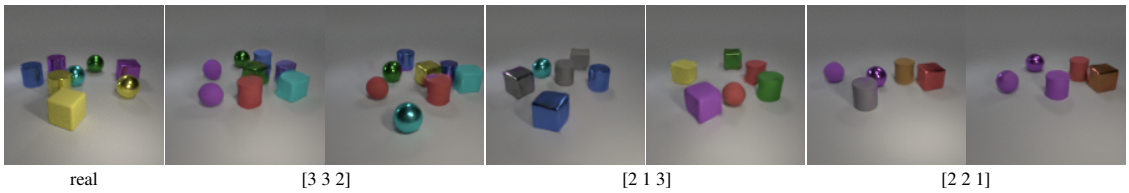
We observed an average count prediction accuracy of 93%, with an individual accuracy of 91% for count one and 90% for count two respectively (a more detailed analysis of



(a) CLEVR-2 and CLEVR-3(simple) - Count vector corresponds to number of cylinders, spheres and cubes.



(b) CLEVR-2 - Count vector corresponds to number of cylinders and spheres.



(c) CLEVR-3 - Count vector corresponds to number of cylinders, spheres and cubes.

(d) SVHN-2 - Count vector corresponds to per digit count.

(e) CityCount - Count vector corresponds to number of cars and persons. Boxes are drawn around objects of interest for ease of visualization.

Figure 3: Generated MC^2-StyleGAN2 images for different count combination across datasets.

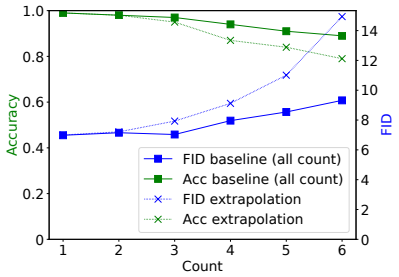
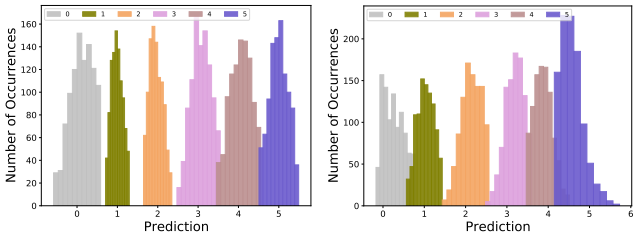


Figure 4: CLEVR-2 extrapolation on spheres based on FID and average count accuracy (Acc). The dotted line indicates the extrapolation performance.

the count prediction on SVHN is provided in the Appendix). We frequently noticed incorrect labels in the original dataset which might affect the count label and prediction accuracy.

4.3. CityCount



(a) Cars. (b) Persons.

Figure 5: Count performance on CityCount generated images. The figure shows the predicted count values for car and person class of the generated samples from our model.

Dataset To evaluate our method on complex real world scenarios, we introduce a count based dataset derived from Cityscapes images, CityCount. The images in CityCount are collected by cropping 256×256 size patches with defined number of *cars* and *persons* from Cityscapes. The dataset contains images with at most five instances from each of these classes and roughly 1000 images per object class count combination. To equip our dataset with additional count information, we determine the number of objects per class in each image from the 2D bounding box information of cars and persons from the Cityscapes-3D [14] and the CityPerson dataset [50]. To allow for more diverse appearances of persons in the training set, classes including *pedestrian*, *sitting person* and *rider* in the Cityscapes images are considered as positive samples when counting the number of persons in the images. This further increases the complexity of the CityCount dataset in terms of spatial arrangement, since the network has to infer a reasonable placement of persons, like pedestrians on the sidewalk and riders on the road. Since such additional spatial constraints are not explicitly specified, this makes our dataset more interesting and challenging for evaluating the proposed approach. Most importantly, the bounding boxes, that were used to generate the training data, were not provided to the model during training.

Evaluation To account for the limited amount of training images, we used the adaptive discriminator augmentation technique [27] for training our model. Samples of real and generated images with their respective count vector are shown in Figure 3e. Each count vector of size two represents the number of cars and persons. For the ease of visualization, boxes are drawn around objects of interest. The model generates images with diverse background and well defined person and car class placed spatially at reasonable locations. As shown in the generated sample of 1 car and 2 persons combination in Figure 3e, the person placed in the road can be seen along with a bike while the second person is standing on the sidewalk. The model learns to distinguish between the pedestrian and the rider class even without an explicit definition of them in the training set.

We evaluate the predictive performance of the count subnetwork for the car and person classes in Figure 5a and 5b respectively. Here, we compare the predicted count values on the generated samples with the true count provided to the generator network during test time. Since in many samples of the training set persons are only partially visible and often out of focus or of low resolution, we observed a comparatively poor count performance for the person class. For higher counts, 4 or 5, the relatively low performance is presumably due to the lower number of training samples and severe occlusions for the corresponding count.

4.4. Ablations

We perform an ablation study on synthetic dataset CLEVR and the real dataset CityCount to verify the importance of the additional count loss, generator design, weight sharing in the discriminator and the conditioning methods.

Count loss We train our model without the count regression network and condition the generator and discriminator with the count label. The rest of our architecture is unchanged. The observed values (w/o count loss in Table 2) show that removing the count loss substantially degrades the performance both in terms of count prediction and image quality.

Generator architecture We consider two different generator configurations introduced in StyleGAN2. One that uses output skip connections and a second one that uses residual connections. As shown in Table 2 (residual and output skip generator), our proposed dense like connections achieves overall good performance in terms of both count prediction and image quality.

Method	Dataset					
	CLEVR-2		CLEVR-3		CityCount	
	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)
w/o Count loss	0.78	18.67	0.80	30.34	0.51	20.24
w/o Discriminator weight sharing	0.91	33.42	0.84	31.03	0.69	15.78
w/o Label mapping	0.90	11.01	0.85	11.32	0.59	8.84
Residual generator	0.94	8.28	**0.93**	11.94	0.65	11.72
Output skip generator	0.94	8.62	0.92	8.98	0.72	10.71
MC^2-StyleGAN2(ours)	**0.95**	**7.98**	0.92	**8.94**	**0.78**	**8.33**

Table 2: Ablation study across datasets based on the Average count accuracy (Acc) and Fréchet Inception Distance (FID). The table shows the validity of the proposed architecture choices in our method.

Weight sharing in the Discriminator We compute the evaluation metrics for our model without weight sharing between the count sub-network and the discriminator. The observed values in Table 3 (w/o discriminator weight sharing) show that the model failed to generate the object count correctly. This confirms the positive impact of weight sharing to regularize the count information and inform the discriminator.

Count conditioning in Generator Lastly, we consider the setting where the count vector is not concatenated to every layer in the mapping network in the generator. The results in Table 3 (w/o label mapping) show that the predictive performance is degraded in this setting. This confirms the benefit of using a count vector based mapping network to propagate the multiple-class count effectively during training.

5. Comparison with other methods

We compare the quantitative performance of other conditional GAN variants, CGAN [32], InfoGAN [7], ACGAN [37] and TACGAN [16], for multiple-class counting on CLEVR and SVHN images. In order to have a fair comparison of our method with these conditional GAN variants, we use a less evolved network architecture in our proposed model. We call this simplified version of our approach, MC^2-SimpleGAN. The MC^2-SimpleGAN generator gets as input a combination of randomly sampled noise and a multiple-class count vector. The generator architecture is inspired by Densenet architecture [21] and includes two dense blocks (where the output from each layer is connected in a feed forward fashion to its succeeding layers) followed by two fully connected layers. The discriminator includes a convolutional based adversarial network and a count regression network with weight sharing. For more architectural details please refer to the Appendix.

The initial results (row 1 to 3 in Table 3) indicate that the considered conditional GAN models did not perform well both in terms of image quality and FID. We even observed mode collapse for CGAN. Hence, we replaced the generator architecture of these models with the a Densenet based generator to improve the performance (rows 4 to 6 in Table 3). Although we could greatly improve the initial performance of these models (which shows the positive impact of the proposed Densenet based generator), MC^2-SimpleGAN clearly outperforms other methods in the envisioned setting. Further the quality of the generated images is improved with the proposed MC^2-StyleGAN2.

6. Training count prediction network using synthetic images

We further demonstrate the usability of the images generated by MC^2-StyleGAN2 for training a count prediction network. In particular, we use a multiple-class extension of regression-based architecture similar to the discriminator of MC^2-SimpleGAN . The network aims to predict the number of objects per object class for the corresponding input images. We design two experiments in this setting using CLEVR and CityCount images. Since the quality of person instances in CityCount images is comparatively low, we also consider a subset of CityCount called CityCar, comprising solely of car class. The average count accuracy of the model is considered as the evaluation metric.

In the first experiment, we evaluate whether the generated images can improve the count performance when combined along with real images during training. For baseline comparison, the count prediction network is initially trained with real images alone (first row in Table 4). The network is then trained with a combination of real and augmented real images (second row in Table 4). The observed count accuracy is then compared with the performance of the network when trained with real and the generated images (third row in Table 4). For fair comparison we consider equal number of augmented and synthetic images. As shown in Table 4 for CLEVR and CityCar images the combination of real and synthetic images (Real+Syn) improved the baseline setting (Real only) and the combination of real and augmented images (Real+Aug). For CityCount, similar count performance is observed for both Real+Aug and Real+Syn.

Method	Dataset					
	CLEVR-2		CLEVR-3		SVHN-2	
	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)	Acc(↑)	FID(↓)
CGAN	0.31	119.23	0.39	186.13	0.39	170.80
InfoGAN	0.37	101.45	0.40	135.36	0.43	151.98s
ACGAN	0.38	99.88	0.40	132.23	0.41	150.56
TACGAN	0.40	92.04	0.42	120.11	0.45	138.29
CGAN(ourG)	0.38	88.79	0.45	152.56	0.55	90.34
InfoGAN (ourG)	0.40	75.23	0.44	112.34	0.55	82.13
ACGAN(ourG)	0.41	55.24	0.42	91.02	0.58	70.28
TACGAN(ourG)	0.44	49.01	0.47	87.64	0.61	65.77
MC^2-SimpleGAN(ours)	0.90	47.95	0.89	85.48	0.92	57.52
MC^2-StyleGAN2(ours)	**0.95**	**7.98**	**0.92**	**8.94**	**0.93**	**10.90**

Table 3: Comparison with other methods across datasets based on the Average count accuracy (Acc) and Fréchet Inception Distance (FID). Underlined values denotes the proposed method performance on simple (MC^2-SimpleGAN) and bold values with complex architecture (MC^2-StyleGAN2).

Training data	Acc(↑)		
	CLEVR	CityCount	CityCar
Real only	0.81	0.68	0.77
Real + Aug	0.81	**0.71**	0.78
Real + Syn(ours)	**0.86**	**0.71**	**0.80**

Table 4: Average count accuracy across datasets for different training data setting.

Training data	Acc(↑)		
	CLEVR	CityCount	CityCar
Real only	0.81	**0.68**	0.75
Syn(ours) only	0.40	0.30	0.39
25% Real only	0.65	0.41	0.59
25% Real + 75% Syn(ours)	0.67	0.45	0.62
50% Real only	0.76	0.56	0.69
50% Real + 50% Syn(ours)	0.81	0.60	0.75
75% Real only	0.77	0.65	0.74
75% Real + 25% Syn(ours)	**0.83**	**0.68**	**0.76**

Table 5: Average count accuracy across datasets when count prediction network trained with real and generated images (Syn) at various proportions.

In the second experiment, we investigate the potential of the generated images to replace the real images during training, without compromising the count accuracy performance. We consider the setting where the network is trained with a combination of real and synthetic images at various ratios. Initially, the network is trained with only real images and then with only synthetic images. We gradually replace the real images with synthetic images at various proportions and evaluate the count performance for each setting as shown in Table 5. For the baseline comparison of each setting, we consider the count accuracy of the network when trained with the corresponding ratio of the real images only (x% Real only in Table 5). As seen in Table 5, 50% of real images could be replaced by the generated images without compromising the overall count performance for both CLEVR and CityCar images. The synthetic images could also improve the overall count performance of the network while replacing 25% of real images for both CLEVR and CityCar images. For CityCount images, 25% of real images could be replaced by the generated images without compromising the overall count performance.

7. Conclusion

In this paper, we investigate the potential of GANs to guide the image generation process based on the number of objects of different classes in the images. While the task of counting is in general very challenging for deep learning approaches, our proposed method can generate images based on the multiple-class count vector in synthetic and real world datasets. Our experiments indicate that the numerosity of objects in the images provides strong information regarding their distinguishability during feature learning and hence allows control of the image generation process. Our evaluation further shows that the model is able to interpolate and extrapolate to unseen counts for specific classes. Even without providing additional information such as the locations of objects in the image, the network infers a reasonable spatial layout and realization of the objects from the training data distribution solely using the count information.

References

[1] V. Agarwal, Rakshith Shetty, and M. Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9687–9695, 2020. 2

[2] Namrata Anand and Possu Huang. Generative modeling for protein structures. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 7494–7505. Curran Associates, Inc., 2018. 1

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 2425–2433, 2015. 2

[4] Yogesh Balaji, Martin Min, Bing Bai, Rama Chellappa, and Hans Graf. Conditional gan with discriminative filter generation for text-to-video synthesis. pages 1995–2001, 08 2019. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In International Conference on Learning Representations (ICLR), 2018. 2

[6] A. B. Chan, Zhang-Sheng John Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–7, 2008. 2

[7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In NIPS, 2016. 2, 7

[8] Yuhua Chen, Feng Shi, Anthony G. Christodoulou, Zhengwei Zhou, Yibin Xie, and Debiao Li. Efficient and accurate mri super-resolution using a generative adversarial network and 3d multi-level densely connected network. In MICCAI, 2018. 3

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 2

[10] S. Dehaene. The number sense: How the mind creates mathematics. Oxford University Press, 2011. 1

[11] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8857–8866, 2018. 2

[12] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), pages 2685–2688, 2012. 2

[13] Giselle Flaccavento, Victor S. Lempitsky, Iestyn Pope, Paul R. Barber, Andrew Zisserman, J. Alison Noble, and Boris Vojnovic. Learning to count cells: Applications to lens-free imaging of large fields. 2011. 2

[14] Nils Gählert, Nicolas Jourdan, Marius Cordts, Uwe Franke, and Joachim Denzler. Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection. CoRR, abs/2006.07864, 2020. 6

[15] Mario Valerio Giuffrida, Hanno Scharr, and Sotirios A. Tsaftaris. Arigan: Synthetic arabidopsis plants using generative adversarial network. In 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), pages 2064–2071, 2017. 3

[16] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxiliary classifiers gan. Advances in neural information processing systems, 32:1328–1337, 12 2019. 2, 7

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press. 2

[18] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In Advances in neural information processing systems (NeurIPS), pages 5767–5777, 2017. 2

[19] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing, 28(11):5464–5478, Nov 2019. 2

[20] Tobias Hinz, Stefan Heinrich, and Stefan Wermter. Generating multiple objects at spatially distinct locations. In International Conference on Learning Representations, 2019. 2

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, 2017. 7

[22] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017. 2

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 5967–5976, 2016. 2

[24] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. pages 1988–1997, 2017. 2, 4

[25] Mingu Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. arXiv: Computer Vision and Pattern Recognition, 2020. 2, 4

[26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017. 2

[27] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In Proc. NeurIPS, 2020. 2, 4, 6

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. arXiv preprint arXiv:1912.04958, 2019. 1, 2, 4

[30] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 1378–1387, New York, New York, USA, 2016. PMLR. 2

[31] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 2

[32] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. 2014. 2, 7

[33] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957, 2018. 2

[34] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957, 2018. 2, 4

[35] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. ArXiv, abs/1802.05637, 2018. 2

[36] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. NIPS, 2011. 2, 4

[37] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In ICML, 2017. 2, 7

[38] Sangeun Oh, Yongsu Jung, Seongsin Kim, Ikjin Lee, and Namwoo Kang. Deep Generative Design: Integration of Topology Optimization and Generative Models. Journal of Mechanical Design, 141(11), 09 2019. 111405. 1

[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015. 2

[40] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015. 2

[41] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In International Conference on Machine Learning (ICML), 2016. 2

[42] Mengye Ren and Richard S. Zemel. End-to-end instance segmentation with recurrent attention. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 293–301, 2017. 2

[43] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. ArXiv, abs/1511.08250, 2016. 2

[44] S. Seguí, O. Pujol, and J. Vitrià. Learning to count with deep object features. In 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 90–96, 2015. 3

[45] Oliver Sidla, Yuriy Lypetskyy, Norbert Brandle, and Stefan Seer. Pedestrian detection and tracking for counting applications in crowded situations. pages 70 – 70, 12 2006. 2

[46] Andrew Trask, Felix Hill, Scott E. Reed, Jack W. Rae, Chris Dyer, and Phil Blunsom. Neural arithmetic logic units. In NeurIPS, 2018. 3

[47] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. CoRR, abs/1502.05698, 2016. 2

[48] Karen Wynn. Children's understanding of counting. Cognition, 36(2):155–193, 1990. 1

[49] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi-Li Zhang, Haibin Lin, Yu e Sun, Tong He, Jonas Mueller, R. Manmatha, Mengnan Li, and Alexander J. Smola. Resnest: Split-attention networks. ArXiv, abs/2004.08955, 2020. 2

[50] Shanshan Zhang, Rodrigo Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4457–4465, 2017. 6

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251, 10 2017. 2