SAMPLING STRATEGIES FOR GAN SYNTHETIC DATA

Binod Bhattarai, Seungryul Baek, Rumeysa Bodur, Tae-Kyun Kim

{b.bhattarai, s.baek15, r.bodur18, tk.kim}@imperial.ac.uk

ABSTRACT

Generative Adversarial Networks (GANs) have been used widely to generate large volumes of synthetic data. This data is being utilised for augmenting with real examples in order to train deep Convolutional Neural Networks (CNNs). Studies have shown that the generated examples lack sufficient realism to train deep CNNs and are poor in diversity. Unlike previous studies of randomly augmenting the synthetic data with real data, we present our simple, effective and easy to implement synthetic data sampling methods to train deep CNNs more efficiently and accurately. To this end, we propose to maximally utilise the parameters learned during training of the GAN itself. These include discriminator's realism confidence score and the confidence on the target label of the synthetic data. In addition to this, we explore reinforcement learning (RL) to automatically search a subset of meaningful synthetic examples from a large pool of GAN synthetic data. We evaluate our method on two challenging face attribute classification data sets viz. AffectNet and CelebA. Our extensive experiments clearly demonstrate the need of sampling synthetic data before augmentation, which also improves the performance of one of the state-of-the-art deep CNNs *in vitro*.

1. INTRODUCTION AND RELATED WORKS

Applications of deep learning algorithms and frameworks in different computer vision tasks, such as image classification [1, 2], face recognition [3, 4, 5], face attribute classification [6, 7, 8, 9, 10] are not new anymore. However, the bottleneck of training these algorithms is the need of large volumes of data and resources, and collecting such large volumes of data is expensive, daunting and requires experts. Some of the tasks such as face recognition, attribute recognition have to face another level of obstacle due to privacy issues.

To tackle such problems, research on augmenting the synthetic data with real data is growing these days [11, 12]. However, the research community is more focused on engineering the architecture of deep networks in comparison to data engineering. There are several network architectures that are being proposed based on AlexNet [2] to Inception Net [13], ResNet [1], to mention a few. In this paper, we propose methods to engineer the training data by discarding unwanted synthetic data before augmenting with real data.

Fig. 1. Real(left) vs Synthetic images (3 right)

One of the most common and successful methods to augment data to train a classification network is applying geometric transformations on images [2], such as rotation, translation, flipping, cropping. Another study [14] identifies the limitations of geometric transformation of not being able to preserve the label (horizontal flip of 6 results into 9 in MNIST data set) of the synthetic data in every case. Thus, feeding such examples during training hurts the performance of the model. To address this issue, [15] recently proposed a method to perform data specific geometric augmentation. Even then, methods of this category still depend on a single input image to generate multiple synthetic images.

Another line of research for data augmentation is the use of large synthetic data generated by GANs [16, 11, 17, 12]. In these methods, synthetic data are used to augment real data randomly when training CNNs. Several GANs [18, 19] are being proposed to generate synthetic examples by translating images from a source category to target categories. Although the photo realism of the synthetic images generated by GANs is improving rapidly, even after augmenting millions of synthetic images, the improvement is still marginal. Recent study on *Seeing is not necessarily believing* [20] observed that even after augmenting visually plausible synthetic examples the performance of the model is degraded. This could be due to large number of synthetic examples not preserving target label. Another study on power of GAN [21] demonstrates that random augmentation of synthetic images are not sufficient to improve the performance. The inception score [22] of images generated by most of the GANs are quite low. This suggests that most of the images do not preserve the target label and also lack realism. Due to these shortcomings of synthetic data, it is not useful to feed in all the synthetic examples to train CNNs.

This project is partly funded by EPSRC FACER2VM project

We are interested in mitigating the above mentioned challenges on synthetic data from GANs and maximise their benefits without using any external supervisions. Our methods are less demanding since we are mostly relying on the information, which is available on the GAN itself and do not need additional annotations/source of information. One of them is target label preserving confidence score of synthetic examples, which is easy to compute from a pre-trained classifier on limited real examples. Another one is the confidence score on realism of synthetic data, which can be easily computed from the discriminator. Finally, we propose to learn policies to augment or not to augment the synthetic data using RL algorithm. This type of algorithms are successful when some of the functions in the pipelines are non-differential.

We validate our method on two different challenging face attribute classification data sets viz. CelebA [6] and AffectNet [23]. We use StarGAN [18], one of the state-of-the-art face attribute translation GANs. We performed extensive experiments to validate our idea. To the best of our knowledge, this is the first work to do such systematic study on selecting the useful synthetic data from a pool of millions of synthetic data.

Our contributions can be summarised as: i) proposed two different efficient, effective and easy to implement data sampling methods, ii) applied RL algorithm for sub-sampling GAN synthetic data, iii) performed extensive systematic empirical experiments demonstrating the need of sub-sampling meaningful data and improved the performance of state-of-the art deep architecture *in vitro*.

2. PROPOSED METHOD

In this section, we describe our proposed methods in a detailed level. Given real data $D_r = \{(x_i^r, y_i^r)\}_{i=1}^{i=L}, (x^r, y^r) \sim p_r(X^r, Y^r; \theta^r)$ and synthetic data $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{i=M}, (x_s, y_s) \sim p_s(X_s, Y_s; \theta_g)$ we are interested in sub-sampling the synthetic data. Here, θ_g is the parameter of the generator and θ_r is the distribution of the real data set, which is known only empirically. We have a scenario where $L < M$ and our objective is to select N number of synthetic examples s.t. $N << M$ and augment with the real data set $\{(x_i^r, y_i^r)\}_{i=1}^{i=L} \bigcup \{(x_i^s, y_i^s)\}_{i=1}^{i=N}$ to train a model. It is important that we will improve the performance of the classifier on real validation data set.

Fig. 2 shows the schematic diagram of our proposed pipelines. First, synthetic data generated by the generator is passed through the data-sampler. There are three different types of data sampling techniques based on class conditional confidence score, realism conditional score and reinforcement learning. In this work, we are evaluating one sub-sampler at a time. This sub-sampler discards the unwanted data and lets only pass the useful data points. The filtered synthetic data is augmented with the real data set and then we train the classifier. The volume of the data set, which is discarded is comparatively larger, in the order of few folds, in comparison to the passed data to train the final classifier. We discuss about the sub-sampling functions and the generator in the following sub-sections.

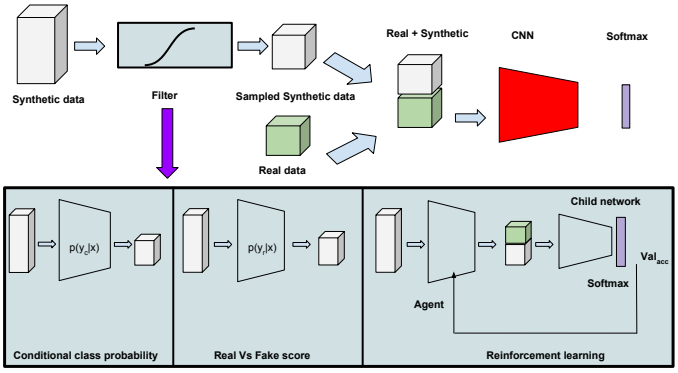

Fig. 2. Schematic diagram of the proposed method. We propose to have three types of filters (Conditional class probability(cl-sam), Real vs. Fake score (cr-sam) and Reinforcement learning(RL)) to get rid of unwanted synthetic data.

Generator: We employed StarGAN [18] as our generator. To reiterate, our methods are generic and can be used with any other types of GANs or generators. StarGAN takes the source image, and target label as input and returns the translated image. For attributes synthesis, we used the publicly available pre-trained model, whereas for expression synthesis, we used training data from AffectNet, one of the largest data sets annotated with different expressions.

Class conditional probability (cl-sam): We propose to use class conditional probability, which is commonly known as class confidence score, as one of the filters to discard the unwanted examples. For a given synthetic example, we computed class conditional probability $P(y_c|x_s; \theta_c)$. Here, x_s represents the synthetic data, y_c the target class c and θ_c the model parameters of the classifier pre-trained on real data only. This confidence score is utilized to filter out the synthetic examples. We rank the synthetic examples based on the conditional target class label(on descending order) and select the top-K. We called the sampler based on this score as *cl-sam*.

Realism conditional probability (rl-sam): We propose to use the confidence on realism as another parameter for our sampling function. We use parameters of the discriminator to compute the realism confidence score on synthetic data. We then rank them (in descending order) for each category. The top-K are selected to augment the real training data set. We called the sampler based on this score as *rl-sam*.

Reinforcement Learning: We choose a subset of the real training data 1% and select $8\times$ large synthetic data. We assume a scenario where synthetic data is abundant and real data is limited. We train the policy network of 3 (a CNN with 3 convolutional layers and 2 fully connected layers) to

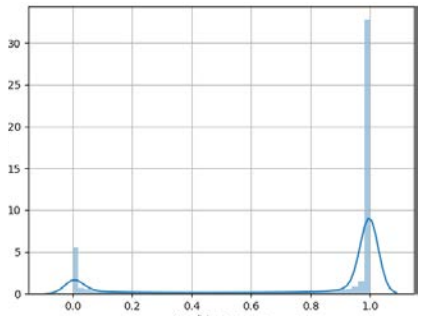

Fig. 3. Distribution of confidence score on AffectNet synthetic data with target label Contempt . X-axis represents the confidence score and Y-axis represents the distribution of Data (in %).

sub-sample the synthetic examples. Our policy network takes image as input, thus the policies are conditioned on the content of the images (this is the main difference from [15]). Fig. 2 shows the schematic diagram of the proposed method. We use the actor and critic method similar to [15] to learn the augmentation policies.

Reward. We compute the reward based on the score on validation set using the child network, similar to [15]. The child network is a small classification network, which mimics the final classification network. The architecture is set same as the aforementioned policy network. We compare the val score with the threshold score. We compute threshold by averaging the val scores in the sliding window of last 5 episodes. If the score is higher than the threshold, we assign +1 to policy, otherwise -1.

3. EXPERIMENTS

3.1. Data sets and their pre-processing

CelebA. This is one of the largest and most widely used data sets for attribute classification. This data set consists of 200K annotated examples and is divided into training, validation and testing set of sizes of 160K, 20K and 20K, respectively. There are 40 attributions in total. For our experiments, we selected 5 important attributes (black hair, brown hair, blonde hair, young and gender) .

AffectNet. This is one of the largest data sets for expressions, emotions and valence arousal estimations. In this data set, there are nearly 1M samples where 400K of them are manually annotated and the rest is automatically annotated. We choose the down sampled version of manually annotated examples for our purpose. This version consists of $88K$ annotated examples. The images are annotated with 8 expressions and split into train, validation and test set.

Synthetic data. We use StarGAN [18], one of state-of-the-art GANs, to generate the synthetic examples. For CelebA, we use publicly available pre-trained model, whereas for AffectNet, we use the training set to train StarGAN from scratch.

Architecture	Resolution	Mean. Acc.	Aug.	Type
AlexNet [23]	$224 \times 224 \times 3$	**50.0**	$0\times$	No aug.
ResNet-50	$64 \times 64 \times 3$	46.1	$0\times$	No aug.
ResNet-50	$128 \times 128 \times 3$	49.6	$0\times$	No aug.
ResNet-50	$128 \times 128 \times 3$	50.3	$1\times$	Random
ResNet-50	$128 \times 128 \times 3$	51.7	$1\times$	cl-sam
ResNet-50	$128 \times 128 \times 3$	**52.2**	$1\times$	cr-sam
ResNet-50	$128 \times 128 \times 3$	52.3	$2\times$	Random
ResNet-50	$128 \times 128 \times 3$	**52.6**	$2\times$	cl-sam
ResNet-50	$128 \times 128 \times 3$	50.9	$2\times$	cr-sam
ResNet-50	$128 \times 128 \times 3$	51.0	$5\times$	Random
ResNet-50	$128 \times 128 \times 3$	**52.2**	$5\times$	cl-sam
ResNet-50	$128 \times 128 \times 3$	51.7	$5\times$	cr-sam
ResNet-50	$128 \times 128 \times 3$	51.8	$2.6\times$	RL

Table 1. Comparison of mean average performance our evaluated methods with existing art on AffectNet.

We generated synthetic data up to 12-folds and 7-folds of real data for CelebA and AffectNet, respectively.

We train the CNN at different resolutions. Tab 1 shows the baseline performance on AffectNet at different resolutions. As we observe that $128 \times 128 \times 3$ attains the performance of previous method reported on [23] with the resolution of $224 \times 224 \times 3$, we set this resolution for further evaluations. For CelebA, we choose $64 \times 64 \times 3$ to reduce the computing complexity, since this data set is comparatively larger. We resize CelebA and AffectNet to $72 \times 72 \times 3$ and $144 \times 144 \times 3$, respectively and randomly crop on 4 corners and centre. We also randomly flip the images when training the network, while at test time, we centre crop the images.

3.2. Evaluation Protocol and Compared Methods

We compute attribute classification accuracy on two benchmark data sets: CelebA and AffectNet data set to evaluate the proposed methods quantitatively. We also provide qualitative visualisations to compare the quality of the images sampled by the evaluated methods. We evaluated 4 different augmentation methods including baseline on state-of-the-art ResNet-50 deep architecture.

Random augmentation. This is the most commonly used augmentation technique. We randomly sub-sample the synthetic set on different proportions $(1\times, 2\times, 5\times)$ compared to real data. We augment this data with real data to re-train the CNN.

Conditional class conf. sampler (cl-sam) We select top-K (where $K = 1\times, 2\times, 5\times$) of the synthetic data ranked on the basis of class-conditional confidence score and augment them with real data.

Discriminator Real/Fake score-based sampler (cr-sam). We select top-K (where $K = 1\times, 2\times, 5\times$) of the synthetic data ranked on the basis of realism vs fake score from discriminator and augment them with real data.

Reinforcement learning (RL)-based sampler. As discussed before, we train the agent to select the useful synthetic data. This agent is applied to the whole synthetic data and only the synthetic data chosen by the agent is used for augmentation.

3.3. Experimental results

Tab.1 shows the comparison of mean accuracy of the proposed method and existing art on AffectNet. From the Tab., augmenting real data with $1\times$ of synthetic data improved the performance in all the cases. However, we observe a difference in performance gain between the evaluated methods. Random augmentation yields the minimum gain ($+0.4\%$) whilst *rl-sam* (based on realism) yields the highest gain ($+2.6\%$). Similarly, *cl-sam* improved the performance by $+2.1\%$. This is expected as the random method is equally likely to select both useful and misleading examples while *rl-sam* and *cl-sam* manage to collect examples that are more realistic and preserve the class-conditional label more confidently, respectively. On further increasing the volume of synthetic data, we observe further improvement on the performance of random and *cl-sam* while the performance of *cr-sam* is slightly degraded. It is because being real does not ensure that the target label of the synthetic data is preserved. The ratio of the performance improvement from $1\times$ to $2\times$ augmentation was lower than when augmentation is of size $1\times$. On further increasing the augmentation size to $5\times$, we observe degradation of the performance of all the three methods in comparison to that of $2\times$. The performance of *cl-sam* is degraded by a minimum margin while the degradation of performance by random sampling is maximum. This supports the fact that there are only a limited number of useful data to augment. In addition, when the size of augmentation increases, the ratio of useful synthetic data to misleading data decreases. Finally, we applied our RL policy to sub-sample the synthetic data. It selected only $2.6\times$ of real data of synthetic data from the whole synthetic data, which is $7\times$ of real data. The performance of RL is slightly lower in comparison to *cl-sam*. However, it outperforms the performance of the other two methods. As we know, *cl-sam* was trained with a real training set of size 88K data to learn the parameters, whereas RL uses no such annotations but learns only from experience. Another potential reason for RL not being as competitive as *cl-sam* is due to huge difference in architecture of child network and final classification network. In our case, as mentioned before, the child network has comparatively much fewer parameters and a different architecture than ResNet-50. Thus, the policies learned for child network may not be necessarily generalised to large classification network. It will be computationally very expensive to have a child network with the parameters similar to that of ResNet-50.

We also evaluated the category level of expression accuracy of the baseline and the proposed methods. Tab. 3 shows the categorical performance comparison on AffectNet. We can see that random augmentation suffers in wide range of performance gain and drop. For example, recognition of *Contempt* improves from 72.2% to 90.2% when the augmentation size is increased from $1\times$ to $5\times$ whereas the recognition of *Sadnees* drops from 60.9% to 46.5%. We did not observe such fluctuations on other compared methods. This suggests that the model trained with randomly augmented data is less stable than other approaches.

Architecture	Resolution	Mean. Acc.	Aug.	Type
[24]	$224 \times 224 \times 3$	80.1	$0\times$	No aug.
[6]	$224 \times 224 \times 3$	87.3	$0\times$	No aug.
[25]	$224 \times 224 \times 3$	88.7	$0\times$	No aug.
[9]	$224 \times 224 \times 3$	91.2	$0\times$	No aug.
[26]	$224 \times 224 \times 3$	91.6	$0\times$	No aug.
ResNet-50	$64 \times 64 \times 3$	90.3	$0\times$	No aug.
ResNet-50	$64 \times 64 \times 3$	91.0	$5\times$	Random
ResNet-50	$64 \times 64 \times 3$	91.1	$5\times$	cl-sam.
ResNet-50	$64 \times 64 \times 3$	91.0	$5\times$	cr-sam.

Table 2. Comparison of mean average performance of our evaluated methods with existing art on CelebA.

		Expressions							
		Anger	Contempt	Disgust	Fear	Happy	Neutral	Sadness	Surprise
$0\times$	Real	41.4	62.5	64.5	68.7	55.0	39.2	50.6	44.0
$1\times$	Random	36.6	**72.2**	69.3	**77.3**	**60.9**	36.7	**60.9**	51.6
	cl-sam	46.6	60.4	68.9	67.3	55.2	40.0	46.8	50.2
	cr-sam	**47.2**	68.8	**77.8**	66.3	55.7	**40.2**	52.7	47.8
$2\times$	Random	46.5	68.4	66.2	60.7	**62.5**	39.9	50.2	46.6
	cl-sam	**48.9**	71.5	75.6	71.6	56.3	**40.5**	51.0	**48.9**
	cr-sam	47.3	**74.8**	**78.9**	**71.7**	55.7	36.1	**65.4**	43.8
$5\times$	Random	44.7	**90.2**	**74.4**	69.0	62.4	**39.6**	46.5	45.3
	cl-sam	**51.0**	63.6	68.4	**70.8**	53.6	38.4	46.5	**51.6**
	cr-sam	49.0	64.3	67.5	67.5	**62.9**	35.4	**49.5**	46.4
2.6x	RL	44.0	71.2	76.2	68.8	60.5	37.0	56.7	48.6

Table 3. Comparison of categorical performances of our evaluated methods on AffectNet.

Similarly, we also evaluated our methods on CelebA, another popular data set for attribute classification. We observe similar trends that we observed on Affectnet. Please refer to Tab. 2 for more details. As in Affectnet, *cl-sam* outperforms other compared methods. We further compared our performance with several state-of-the-art methods. Even though we performed our experiments on $4\times$ lower resolution ($64\times64\times3$) than previous arts, our methods are either outperforming or competitive.

4. CONCLUSIONS

We evaluated three different methods over commonly used augmentation techniques. We propose to use confidence score and realism score based sampler to find a meaningful subset. Finally, we explored RL based sampler, which learns from experiences. From our extensive experiments, we observed that these three techniques outperform the commonly used random augmentation technique. Among these three, we observed that the class conditional based sampler performs the best followed by RL and realism conditional probability based sampler. Each method has its own shortcomings and advantages. Confidence score based sampler requires real training examples. Although realism score based sampler does not require labelled training example, it does not guarantee to preserve the class conditional probability. RL does not require training examples, but it is expensive to train.

5. REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[3] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.

[5] Binod Bhattarai, Gaurav Sharma, and Frederic Jurie, "CP-mtML: Coupled projection multi-task metric learning for large scale face retrieval," in *CVPR*, 2016.

[6] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.

[7] Sunghun Kang, Donghoon Lee, and Chang D Yoo, "Face attribute classification using attribute-aware correlation map and gated convolutional neural networks," in *ICIP*, 2015.

[8] Emily M Hand and Rama Chellappa, "Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification," in *AAAI*, 2017.

[9] Mahdi M Kalayeh, Boqing Gong, and Mubarak Shah, "Improving facial attribute prediction using semantic segmentation," in *CVPR*, 2017.

[10] Binod Bhattarai, Gaurav Sharma, and Frédéric Jurie, "Deep fusion of visual signatures for client-server facial analysis," in *ICVGIP*, 2016.

[11] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb, "Learning from simulated and unsupervised images through adversarial training," in *CVPR*, 2017.

[12] Baris Gecer, Binod Bhattarai, Josef Kittler, and Tae-Kyun Kim, "Semi-supervised adversarial learning to generate photorealistic face images of new identities from 3d morphable model," in *ECCV*, 2018.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[14] Søren Hauberg, Oren Freifeld, Anders Boesen Lindbo Larsen, John Fisher, and Lars Hansen, "Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation," in *AISTATS*, 2016.

[15] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le, "Autoaugment: Learning augmentation policies from data," *CVPR*, 2019.

[16] S. Baek, K. I. Kim, and T-K. Kim, "Augmented skeleton space transfer for depth-based hand pose estimation," in *CVPR*, 2018.

[17] Zhedong Zheng, Liang Zheng, and Yi Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *ICCV*, 2017.

[18] Y. Choi, M. Choi, M. Kim, J-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018.

[19] G. Zhang, M. Kan, S. Shan, and X. Chen, "Generative adversarial network with spatial attention for face attribute editing," in *ECCV*, 2018.

[20] Suman Ravuri and Oriol Vinyals, "Seeing is not necessarily believing: Limitations of biggans for data augmentation," in *ICLR*, 2019.

[21] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari, "How good is my gan?," in *ECCV*, 2018.

[22] Shane Barratt and Rishi Sharma, "A note on the inception score," *arXiv preprint arXiv:1801.01973*, 2018.

[23] Mollahosseini and et al., "Affectnet: A database for facial expression, valence, and arousal computing in the wild," *arXiv preprint arXiv:1708.03985*, 2017.

[24] N. Kumar, P. Belhumeur, and S. Nayar, "Facetracer: A search engine for large collections of images with faces," in *ECCV*, 2008.

[25] J. Wang, Y. Cheng, and R. S. Feris, "Walk and learn: Facial attribute representation learning from egocentric video and contextual data," in *CVPR*, 2016.

[26] Yuechuan Sun and Jun Yu, "Deep facial attribute detection in the wild: From general to specific," in *BMVC*, 2018.