Sampling Strategies for GAN Synthetic Data

Binod Bhattarai, Seungryul Baek, Rumeysa Bodur, Tae-Kyun Kim
Imperial College London
{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 utilized 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 utilize 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 samples 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.

Introduction
Applications of deep learning algorithms and frameworks in different computer vision tasks such as image classification (He et al. 2016; Krizhevsky, Sutskever, and Hinton 2012), face recognition (Taigman et al. 2014; Schroff, Kalenichenko, and Philbin 2015; Bhattarai, Sharma, and Jurie 2016a), face attribute classification (Liu et al. 2015; Bhattarai, Sharma, and Jurie 2016b) are not new anymore. Deep learning algorithms have proven to improve the performance of such applications substantially. 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 etc. have to face another level of obstacle due to privacy issues. Fig. 2 shows the distribution of annotated data from AffectNet (Mollahosseini et al. 2017), which is one of the largest annotated data sets for face attribute classification. We can clearly observe that there are some categories that have an insufficient volume of data to train a deep network optimally.

To tackle such problems, research on augmenting the synthetic data with real data is growing these days (Shrivastava et al. 2017; Gecer et al. 2018). However, the research community is more focused on engineering the architecture of the deep networks in comparison to data engineering. There are several network architectures that are being proposed based on AlexNet (Krizhevsky, Sutskever, and Hinton 2012) to Inception Net (Szegedy et al. 2015), ResNet (He et al. 2016), a few to mention. In this paper, we propose methods to engineer the training data by discarding unwanted synthetic data before augmenting with real data.

One of the most common and successful methods to augment data to train a classification network is applying geometric transformations on images (Krizhevsky, Sutskever, and Hinton 2012), such as rotation, translation, flipping, cropping etc. However, this technique does not guarantee that the label of the synthetic image will be preserved after applying such functions. Another study (Hauberg et al. 2016) identifies the
limitations of geometric transformation of not being able to preserve the label (e.g., 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, et al. (Cubuk et al. 2019) 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 (Baek, Kim, and Kim 2018; Shrivastava et al. 2017; Zheng, Zheng, and Yang 2017; Gecer et al. 2018). In these methods, synthetic data are used to augment real data but randomly when training CNNs. Several GANs (Choi et al. 2018; Zhang et al. 2018) 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 (Ravuri and Vinyals 2019) observed that even after augmenting visually plausible synthetic examples the performance of the model is degraded. This could be due to the large number of synthetic examples not preserving target label. Another study on power of GAN (Shmelkov, Schmid, and Jurie, and Triggs 2006) demonstrates that random augmentation of synthetic images are not sufficient to improve the performance. The inception score (Barratt and Sharma 2018) of images generated by most of the GANs are quite low (See Fig. 1d). This entails, that most of the images do not preserve the target label and also lack realism. Moreover, there is still a domain gap between real data and synthetic data. Some of the research works such as (Shrivastava et al. 2017; Rad, Oberweger, and Lepetit 2018) focus to minimize the domain gap between real and annotated data. However, these methods rely on additional supervision to align the parameters between real domain and target domain. Due to these shortcomings on synthetic data, it is not useful to feed in all the synthetic examples to train CNNs.

We are interested in mitigating the above mentioned challenges on synthetic data from GANs and maximize their benefits without using any external supervisions. Inspired from the success of seminal work on simple yet effective, sampling strategies of bag-of-features for image classification (Nowak, Jurie, and Triggs 2006), we propose two different simple, effective and easy to implement approaches to sub-sample useful synthetic data from a large volume of synthetic data. 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 of the realism of the synthetic data, which can be easily computed from the discriminator. Finally, we propose to learn a policy to augment or not to augment the synthetic data using a reinforcement learning algorithm. Reinforcement Learning algorithms are successful for learning from experiences where there are no annotated examples available.

To validate our ideas we applied our method on two different challenging face attribute classification data sets viz. CelebA (Liu et al. 2015) and AffectNet (Mollahosseini and et al. 2017). We use StarGAN (Choi et al. 2018), which is 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.

We summarize our contributions in the following points:

• Two different efficient, effective and easy to implement data sampling methods
• Applied RL algorithm for sub-sampling GAN synthetic data
• Extensive systematic empirical experiments demonstrating the need of sub-sampling meaningful data.
• Improving the performance of state-of-the art deep architecture in vitro.

Related Works

In this Sec., we further investigate related works that exploit large volumes of synthetic data to augment real data for training their models. Moreover, we will also present some of the works related to face attributes and expressions classifications.

Geometric data augmentation. Data augmentation has been getting popular after the use of CNNs to train a model. This is mainly due to the data voracious nature of CNNs. (Krizhevsky, Sutskever, and Hinton 2012) proposed geometric transformations (random flipping, random cropping etc.) to generate synthetic examples to train their large scale CNN for image classification. Similarly, (Hauberg et al. 2016) proposed to learn data specific geometric transformations to train an image classification network. Recently (Cubuk et al. 2019)
proposed image specific geometric transformation using reinforcement learning algorithms. This methods have improved the performance in comparison to their baselines trained on real data only. (DeVries and Taylor 2017) proposed to do geometric augmentation by adding noise on features, interpolating and extrapolating between features etc. Similarly, (Lemley, Bazrafkan, and Corcoran 2017) proposed a network to automatically generate synthetic image by merging two or more samples from the same class.

**Synthetic model.** Synthetic model suggests an easy way of collecting both 2D images and their corresponding labels, and they have also been used for collecting large-scale database (Varol et al. 2017; Masi et al. 2016). Compared to simple transformations used in data augmentation methods, synthetic models can supply quality data having diverse variations and semantics. However, the issue is the gap between real and synthetic data. Some of the recent works such as (Shrivastava et al. 2017) made attempts to tackle this gap problem.

**Generative Adversarial Networks (GANs).** The generative networks such as GANs and Variational Auto Encoders (VAEs) can be used to generate new samples. Especially, GANs are known to be able to generate realistic samples, while the discriminator and the generator play a “two-player minimax game”. Generating new type data using GANs and augmenting with real data has been investigated in recent works (Baek, Kim, and Kim 2018; Gecer et al. 2018; Zhang et al. 2018; Shmelkov, Schmid, and Alahari 2018; Zhao et al. 2018b; Tran, Yin, and Liu 2017; Zhao et al. 2018a; Huang et al. 2017) and too few to mention. In this paper, we try to investigate methods and tricks to sub-sample instead of randomly augmenting the synthetic images from GAN. Please note that our methods are generic and can be applied for VAEs synthetic data too.

**Reinforcement Learning.** Recently, (Cubuk et al. 2019) applied it for learning automatic polices to find the optimal geometric transformation to generate new examples. However, we applied RL to sub-sample the GAN synthetic data. From our best knowledge, this is the first work to apply RL to sub-sample GAN synthetic data. Another work on learn to simulate (Rui, Shulter, and Chandraker 2019) applied RL to learn the optimal parameters of simulator to generate synthetic data. This method needs to be applied end to end fashion and hence, remains specific to a model. Our method can be applied to any of the pre-trained simulators.

**Face Attributes and Expression Classification** Face attributes and expressions classification is one of the challenging and popular research problems. One of the seminal works on face attributes recognition is from Kumar et al. (Kumar et al. 2009). They propose to learn a classifier for each of the attributes (hair, shape of nose, gender etc.) and use the output of the classifier to encode faces for face verification purposes. (Hand and Chellappa 2017) propose to jointly learn the parameters of the face attributes which share common traits. Similarly, (Rudd, Günther, and Boult 2016) proposed a joint optimisation function to model the parameters of different attributes together. (Liu et al. 2015) proposes a cascade of deep networks to predict attributes of unaligned faces and also proposes CelebA, one of the largest databases for face attributes classification. In (Zhang et al. 2014), a pose-normalized CNN is proposed to estimate the attributes. Similarly, (Kalayeh, Gong, and Shah 2017) proposed to use semantic segmentation as privilege information to train a deep CNN. Recently, (Sun and Yu 2018) proposed to learn hierarchical CNNs for attributes classification. Above all, most of the works are focused on designing the architecture of CNNs. However, our work is focused to design the training data set by selecting the useful ones. We suggest readers to refer to the survey on expression classification (Li and Deng 2018) for more information.

**Proposed Method**

In this section, we describe our proposed methods in a detailed level. Given real data $D_r = \{(x_i^r, y_i^r)\}_{i=1}^N$, $s$-synthetic data $D_s = \{(x_i^s, y_i^s)\}_{i=1}^M$, $(x, y) \sim p_s(X_s, Y_s; \theta_g)$ we are interested in sub-sampling the synthetic data. Here, $\theta_g$ is the parameter of the generator and $\theta_r$ 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 $\{(x_i^r, y_i^r)\}_{i=1}^N \cup \{(x_i^s, y_i^s)\}_{i=1}^N$ to train a model. It is important that we will improve the performance of the classifier on real validation data set.

To sub-sample the examples from synthetic data, we took two different approaches, which are similar to instance re-weighting for domain adaptation problem (Jiang and Zhai 2007). In our case, the instance weighting is in binary fashion : 1 to select augmentation while 0 for discarding the example. We set the threshold using two approaches from our prior knowledge. Without loss of generality, it is very important for the synthetic data to be as realistic as possible and to preserve its target label $i$‘the class conditional probability should be high. However, in reality, there are many synthetic examples that do not preserve class conditional probability and also realism. Fig. 4 shows the distribution of class-confidence score $p(y|x; \theta_c)$ ($\theta_c$ is the classifier model trained on real data set) predicted by the model trained on real data set. We can see that many synthetic examples from each category deviates from high confidence to low confidence. These examples will be misleading to train the model.

Fig. 3 shows the schematic diagram of our proposed pipelines. First, the 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. The sub-sampler discards the unwanted data and lets only pass the useful data points. The filtered synthetic data is then augmented with the real data set and used to 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 elaborate on the size of the discarded volume of synthetic data in the supplementary sections. We discuss about the sub-sampling functions and the generator in the following sub-sections.

**Generator:** We employed StarGAN (Choi et al. 2018) as our
Synthetic data Filter Sampled Synthetic data Real data
Real + Synthetic
CNN
Conditional class probability
$p(y_c|x)$ $p(y_r|x)$
Real Vs Fake score
Child network
Val
Agent
Reinforcement learning
Softmax
Softmax

Figure 3: Schematic diagram of the proposed method. We propose to have three types of filters (e.g. Conditional class probability(cl-sam), Real vs. Fake score (cr-sam) and Reinforcement learning(RL)) to get rid of unwanted synthetic data.

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. In the similar way, it also takes the synthetic image and source label as input and reconstructs the original 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. For expression synthesis, we trained the model from scratch.

**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 $\theta_c$ the model parameters of the classifier 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-$K$. For the parameters of the $\theta_c$, we employ ResNet50 architecture (He et al. 2016) and train it using the stochastic gradient descent (SGD) by minimizing categorical cross-entropy loss:

$$
\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{c,i}^t \log y_{p,i}^t
$$

Once the classifier is trained, we used this classifier to score the synthetic data and rank them in descending order. We then selected the top-$K$ of the synthetic images from each category and augmented with the real data set. We called the sampler based on this score as *cl-sam*.

**Realism conditional probability (rl-sam):** It is equally important that the synthetic examples are as realistic as possible. 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. Similar to the previous one, we train our classifier again with these new training examples. We called the sampler based on this score as *rl-sam*.

**Reinforcement Learning:** We explore using reinforcement learning setup to select the synthetic data, which makes the model more discriminative, and reduces the redundancies and unwanted noisy data.

We choose a subset of the real training data 1% and select 8× 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 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 (Cubuk et al. 2019)). Fig. 3 shows the schematic diagram of the proposed method. We use the actor and critic method similar to (Cubuk et al. 2019) to learn the augmentation policies.

**Reward.** We compute the reward based on the score on validation set using the child network, similar to (Cubuk et al. 2019). 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.

**Experiments**

In this section, we give information about the dataset we use, the pre-processing method, evaluation protocol and compared baselines. Then, we analyze our experimental results.

**Datasets**

**CelebA.** This is one of the largest and most widely used datasets 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 have selected 5 important attributes.

**AffectNet.** This dataset is one of the largest datasets for expression, emotions and valence arousal estimations. In this dataset, there are nearly 1M samples where 400K of them are manually annotated and the rest is automatically annotated. The manually annotated dataset is further divided into up-sampled and down-sampled cases to handle the imbalance number of annotations of different expressions. We choose the sub-sampled one for our evaluation purpose. This version consists of 88K annotated examples. The images are annotated with 8 different expressions and split into train, val and test set. We use these 8 expression annotations for our training and evaluation purpose.

**Synthetic data.** We use StarGAN (Choi et al. 2018), one of state-of-the-arts GANs, to generate the synthetic examples. Note that any kind of GANs could be used here. For CelebA, we use publicly available pre-trained models, whereas for AffectNet, we use the training set to train StarGAN from scratch. We generated synthetic data up to 12-folds and 7-folds of real data for CelebA and AffectNet, respectively.

**Preprocessing of the data**

We train the CNN at different resolutions. Tab 2 shows the baseline performance on AffectNet at different resolutions. As we observe that 128 × 128 × 3 attains the performance of previous method reported on (Mollahosseini and et al. 2017) with the resolution of 224 × 224 × 3, we set this resolution for further evaluations. For CelebA, we choose 64 × 64 × 3 to reduce the computing complexity, since this dataset is comparatively larger. We resize CelebA and AffectNet to 72 × 72 × 3 and 144 × 144 × 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.

**Evaluation protocol**

We compute attribute classification accuracies on two benchmark datasets: CelebA and AffectNet dataset to evaluate the proposed methods for quantitative evaluations. We also provide qualitative visualisations to compare the quality of the images sampled by the evaluated methods.

**Evaluated methods**

We have experimented with 5 baseline methods using the state-of-the-art ResNet-50 architecture.

**Real data set.** This is the most commonly used and successful data augmentation technique to this date. One of the baselines for us is the deep CNN trained on large scale dataset. As we know, CNNs are trained on very large scale data and the performance is near saturation.

**Random augmentation.** We randomly sub-sample the synthetic set on different proportions (1 ×, 2 ×, 5 ×) compared to real data. We augment this data with real data to re-train the CNNs from scratch. In case of AffectNet, we initialized the network with weights of a pre-trained network for ImageNet for all cases.

**Conditional class conf. sampler (cl-sam)** We compute the class confidence scores on synthetic data by the model trained on real dataset only. As we mentioned before, we rank them based on the confidence score in descending order and select top-K (where $K = 1 \times, 2 \times, 5 \times$) of the synthetic data. We then augment this synthetic data to train the CNN from the beginning.

**Discriminator Real/Fake score-based sampler (cr-sam).** For each synthetic example translated to target category, we computed the discriminator’s real vs fake score. Similar to the confidence score set-up, we re-ranked the synthetic data and selected the top-K examples to augment the real dataset.

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

**Experimental results**

We performed extensive evaluations on proposed methods to validate the ideas. We have compared our methods on two challenging face attributes classification and expression classifications data set. In the following sub-section, we analyse our results in detail. We will first start with quantitative analysis followed by qualitative analysis.

**Quantitative analysis**

**Baseline** Tab 2 shows the mean accuracies of the compared methods and existing art on AffectNet.

From Tab. we can observe that the performance of our baseline implementation ResNet-50 on (128 × 128 × 3) is slightly lower than existing art (−0.4%). We set this architecture and resolution as our baseline and performed further analysis.

**Augmentation** We then augmented real data with different proportions sampled by the evaluated methods. As we expected, the performance on the test set improves as the augmentation size is increased from 0 × to 1 × in all 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 (+2.0%). Similarly, cl-sam observed the gain of (+2.1%). This is expected, as the random method samples both useful and and misleading examples, while rl-sam and cl-sam manage to collect examples that are more realistic and preserve the class-conditional label, respectively. On further
increasing the volume of synthetic data we observe further improvement on the performance of random and \textit{cl-sam}, while the performance of \textit{cr-sam} is slightly degraded. It is because being real does not ensure 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 \(2 \times \) augmentation. However, the performance of \textit{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. With increase in size of augmentation, the ratio of useful synthetic data to misleading data keeps on decreasing. This trend of performance is further supported by Fig. 4. In the Fig., we can clearly see a large number of misleading examples \(i^e\) with low class confidence score. There is similar trend on realism score too. Please refer supplementary material for more details. In addition to this, we also applied our RL policy to sub-sample the synthetic data. It selected only \(2.6 \times \) of real data of synthetic data \((7 \times \) of real data is the size of full synthetic data). The performance of the RL in comparison to \textit{cl-sam} is slightly lower. However, it outperforms the performance of the other two methods. As we know, \textit{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 \textit{cl-sam} is due to huge difference in architecture of child network and final classification network. For us, child network has comparatively very less parameters and different architecture. As we mentioned before, our child network has \(3 \) Conv layers and \(2 \) fully connected layers. Whereas, our classification network is Resnet50. Thus, the policies learned for child network may not be necessarily generalisable to large classification network. It will be computationally highly expensive to have a child network with the parameters similar to that of Resnet50.

We also compared category level of expression accuracy of the compared methods. Tab. 3 shows the categorical performance comparison between all the compared methods on AffectNet. We can see that random augmentation suffers in wide range of performance gain and drop. For example, \textit{Contempt} improves from \(72.2\%\) to \(90.2\%\) when the augmentation size is increased from \(1 \times \) to \(5 \times \). In the similar range, the performance of \textit{Sadness} drops from \(60.9\%\) to \(46.5\%\). We did not observe such trends on other compared methods. This suggests that the model trained with randomly augmented data are less robust to other approaches.

Similarly, we also performed extensive experiments on CelebA, another challenging and widely used data sets for the attributes classification. We observe the similar trends that we observe on AffectNet. Please refer Tab. 1 for more details. As We observed on Affectnet, \textit{cl-sam} is outperforming other compared method. We also compared our performance with several state-of-the-arts method. Even though we performed our experiments on \(4 \times \) lower resolution \(i^e(64 \times 64 \times 3)\) than compared arts, our methods are either outperforming or competitive. Similarly, Tab. 4 shows the categorical attribute classification performance on CelebA.

### Table 1: Comparison of mean average performance our evaluated methods with existing art on CelebA.
In this sub-section we present our qualitative evaluations. In Fig. 5, left block shows the synthetic images from AffectNet while right block shows the CelebA synthetic images. In each block, each column shows a sample from each of the categories of expressions (AffectNet) and attributes (CelebA). While each row shows (from top to bottom) the samples having high/low Class conditional probability and Real vs. Fake score. We can clearly observe that images with high confidence scores are more visually plausible and appealing than low confidence scores. Similarly, images with high realism scores are better in quality than lower realism scores. This further supports our argument and also our empirical evidences.

Conclusions

In this paper we evaluated three different data augmentation techniques over random augmentation technique. Firstly, we propose to use confidence score based sampler to find a
meaningful sub-set. Similarly, we proposed to use realism conditional probability based sampler. Finally, we explored reinforcement learning based sampler, which learns from the experiences. From our extensive experiments, we observed that these three techniques outperform the commonly used random augmentation technique and improves the performance of state-of-the-art CNNs. 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 scored based sampler requires real training examples. Although realism conditional 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 computationally expensive.

Acknowledgements

This work is partly supported by EPSRC FACER2VM project.

References

Baek, S.; Kim, K. I.; and Kim, T.-K. 2018. Augmented skeleton space transfer for depth-based hand pose estimation. In CVPR.

Barratt, S., and Sharma, R. 2018. A note on the inception score. arXiv preprint arXiv:1801.01973.

Bhattarai, B.; Sharma, G.; and Jurie, F. 2016a. CP-mtML: Coupled projection multi-task metric learning for large scale face retrieval. In CVPR.

Bhattarai, B.; Sharma, G.; and Jurie, F. 2016b. Deep fusion of visual signatures for client-server facial analysis. In ICGVIP.

Choi, Y.; Choi, M.; Kim, M.; Ha, J.-W.; Kim, S.; and Choo, J. 2018. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In CVPR.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning augmentation policies from data. CVPR.

DeVries, T., and Taylor, G. W. 2017. Dataset augmentation in feature space. In ICLR Workshops.

Gecer, B.; Bhattarai, B.; Kittler, J.; and Kim, T.-K. 2018. Semi-supervised adversarial learning to generate photorealistic face images from new identities from 3d morphable model. In ECCV.

Günther, M.; Rozsa, A.; and Boulton, T. E. 2017. Affact-alignment free facial attribute classification technique. In IJCB.

Hand, E. M., and Chellappa, R. 2017. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. In AAAI.

Haugberg, S.; Freifeld, O.; Larsen, A. B. L.; Fisher, J.; and Hansen, L. 2016. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In AISTATS.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In CVPR.

Huang, R.; Zhang, S.; Li, T.; and He, R. 2017. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In ICCV.

Jiang, J., and Zhai, C. 2007. Instance weighting for domain adaptation in nlp. In ACL.

Kalayeh, M. M.; Gong, B.; and Shah, M. 2017. Improving facial attribute prediction using semantic segmentation. In CVPR.

Kang, S.; Lee, D.; and Yoo, C. D. 2015. Face attribute classification using attribute-aware correlation map and gated convolutional neural networks. In ICIP.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In NIPS.

Kumar, N.; Belhumeur, P.; and Nayar, S. 2008. Facetracer: A search engine for large collections of images with faces. In ECCV.

Kumar, N.; Berg, A. C.; Belhumeur, P. N.; and Nayar, S. K. 2009. Attribute and simile classifiers for face verification. In ICCV.

Lemley, J.; Bazrafkan, S.; and Corcoran, P. 2017. Smart augmentation learning an optimal data augmentation strategy. IEEE Access.

Li, S., and Deng, W. 2018. Deep facial expression recognition: A survey. arXiv preprint arXiv:1804.08348.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In ICCV.

Masi, I.; Tran, A. T.; Hassner, T.; Leksut, J. T.; and Medioni, G. 2016. Do we really need to collect millions of faces for effective face recognition? In ECCV.

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

Nowak, E.; Jurie, F.; and Triggs, B. 2006. Sampling strategies for bag-of-features image classification. In ECCV.

Rad, M.; Oberweger, M.; and Lepetit, V. 2018. Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In CVPR.

Ravuri, S., and Vinyals, O. 2019. Seeing is not necessarily believing: Limitations of biggans for data augmentation. In ICLR.

Rudd, E. M.; Günther, M.; and Boult, T. E. 2016. Moon: A mixed objective optimization network for the recognition of facial attributes. In ECCV.

Ruiz, N.; Schulter, S.; and Chandraker, M. 2019. Learn to simulate. In ICLR.

Schoff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In CVPR.

Shmelkov, K.; Schmid, C.; and Alahari, K. 2018. How good is my gan? In ECCV.

Shrivastava, A.; Pfister, T.; Tuzel, O.; Susskind, J.; Wang, W.; and Webb, R. 2017. Learning from simulated and unsupervised images through adversarial training. In CVPR.
Sun, Y., and Yu, J. 2018. Deep facial attribute detection in the wild: From general to specific. In BMVC.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In CVPR.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In CVPR.

Tran, L.; Yin, X.; and Liu, X. 2017. Disentangled representation learning gan for pose-invariant face recognition. In CVPR.

Varol, G.; Romero, J.; Martin, X.; Mahmood, N.; Black, M. J.; Laptev, I.; and Schmid, C. 2017. Learning from synthetic humans. In CVPR.

Wang, J.; Cheng, Y.; and Feris, R. S. 2016. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In CVPR.

Zhang, N.; Paluri, M.; Ranzato, M.; Darrell, T.; and Bourdev, L. 2014. Panda: Pose aligned networks for deep attribute modeling. In CVPR.

Zheng, Z.; Zheng, L.; and Yang, Y. 2017. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In ICCV.