DiMViDA: Diffusion-based Multi-View Data Augmentation

DIMVIDA: Diffusion-based Multi-View Data Augmentation

G. Di Giacomo[1], G. Franzese[2], T. Cerquitelli[1], C. F. Chiasserini[1,3,4], P. Michiardi[2]
1: Politecnico di Torino, Italy – 2: EURECOM, France – 3: CNR-IEIIT, Italy – 4: CNIT, Italy

***Abstract*—We present DIMVIDA, a Diffusion-based Multi-View Data Augmentation method built upon an innovative approach for Novel View Synthesis, which uses an extension of diffusion generative models that accepts any number of input views and that can generate any number of missing output views. In this work, our goal is to analyze the benefits of such a generative model in the context of object classification. Given a single input view, we compare the object classification performance of state-of-the-art models, namely ResNet18 and MobileNetV3, using the input view, versus its application to novel views synthesized by our generative model, using such synthetic views to augment the training set. Notably, differently from other works, we also adopt such a multi-view data augmentation method at inference. Our experimental findings illustrate that novel view synthesis can enhance object classification capabilities.**

***Index Terms*—ML as a Service, DNN training, edge computing, energy-aware models for learning**

I. INTRODUCTION

Generative modeling is an increasingly important research area, as it enables the learning of data distribution and the generation of synthetic samples that mimic real-world data. In recent years, diffusion models [1]–[5] have emerged as the state-of-the-art models for image generation, outperforming Generative Adversarial Networks (GANs) [6] and Variational AutoEncoders (VAEs) [7].

Among the possible applications, data augmentation strategies based both on GANs [8], [9] and diffusion models [10] have been adopted to produce synthetic data that are then used to enhance the capabilities of trained Deep Neural Networks (DNN). For example, while standard data augmentation methods are based on color (e.g. change of brightness, contrast, saturation) and geometric (e.g., flipping, rotation) transformations, [11] leverages diffusion models to perform data augmentation by changing the semantics of the data, increasing their diversity.

Recently, there has been a growing interest in multi-modal [12], [13] and multi-view [14]–[16] diffusion models, which address the challenge of modeling multiple inputs representing the same concept using different modalities, e.g., image and text, or different views. In this work, we present DIMVIDA, a Diffusion-based Multi-View Data Augmentation method that uses a Novel View Synthesis diffusion model to augment both the training and the evaluation pipeline in the context of a classification task.

Specifically, we consider a single-view dataset consisting of N samples, each representing a view of an object, and we augment it by generating for each sample three additional images of the same object from different viewpoints. To do so, we employ the latent diffusion model (LDM) introduced in [12], which applies multi-time masked diffusion to endow the model with conditional generation capability. Then, the resulting augmented set is used to train a classification model. At inference, we follow the same augmentation procedure also for the test set, producing a set of 4 views for each test sample. Next, given a set of multi-view images, we evaluate the classification model individually for each view and, finally, we compute the mean of the 4 outputs produced by the classification model to predict the samples' class. We corroborate the effectiveness of our method by performing experiments on a multi-view dataset using two state-of-the-art classification models, i.e., ResNet18 [17] and MobileNetV3 [18], both training the models from scratch and fine-tuning them starting from pre-trained weights. Our approach leads to improvements in classification accuracy up to 20%.

A similar approach was introduced in [19]; however, a GAN was used instead of a diffusion model and, remarkably, data augmentation is only carried out to extend the training set. In contrast, in our work, we also augment the inference pipeline, enabling us to achieve superior performance compared to only augmenting the training set.

The rest of the paper is organized as follows. Sec. II introduces the latent diffusion model utilized to generate novel synthetic views, while Sec. III presents the designed methodology. The experimental details and performance of DIMVIDA are presented in Sec. IV, while Sec. V discusses some related work. Finally, Sec. VI draws our conclusions.

II. LATENT DIFFUSION MODEL

In this section, we summarize the architecture of our Novel View Synthesis generative model, namely a latent diffusion model, which is composed of two building blocks: a deterministic autoencoder and a score-based diffusion model.

A. Autoencoder

The deterministic autoencoder consists of two parts, the encoder e_ϕ and the decoder d_ψ, and is trained independently and prior to the diffusion model. The autoencoder is a model that learns to reconstruct its input. Specifically, the encoder maps the input in a lower-dimensional space, inducing a latent variable, while the decoder maps back such latent variables to the input space, producing an output that should be as close to the input as possible.

Formally, given a data distribution $p(x)$, the autoencoder is trained by minimizing the following objective function:

$$\mathcal{L} = \int p(x) l(x - d_\psi(e_\phi(x)))\, dx\,. \tag{1}$$

where l is the desired distance function.

B. Score-based diffusion model

After training the deterministic autoencoder, the encoder is used to produce latent representations of the data, which are then used to train the score-based diffusion model. At this stage, the diffusion model learns the latent distribution, enabling conditional generation during inference.

The score-based diffusion model involves two steps, namely, the forward and the backward diffusion processes. The forward process is a stochastic noising process injecting noise into the input data, i.e., the latent representations, and is defined by the following Stochastic Differential Equation (SDE):

$$\mathrm{d}R_t = \alpha(t)R_t\mathrm{d}t + g(t)\mathrm{d}W_t,\;\; R_0 = Z \sim q(r,0), \tag{2}$$

where $\alpha(t)R_t$ and $g(t)$ are the drift and diffusion terms, respectively. W_t is a Wiener process, while $q(r,t)$ denotes the time-varying probability density of the stochastic process at time $t{\in}[0,T]$, with finite T and initial conditions $q(r,0){=}q_\phi(r)$.

To generate a new sample, we need to reverse the noising process by simulating the reverse-time SDE:

$$\begin{aligned}\mathrm{d}R_t{=}\left(-\alpha(T{-}t)R_t{+}g^2(T{-}t)\nabla\log(q(R_t,T{-}t))\right)\mathrm{d}t\\ {+}g(T{-}t)\mathrm{d}W_t,\;\; R_0{\sim}q(r,T).\end{aligned} \tag{3}$$

To solve (3), a parametric score network $s_\chi(r,t)$ is used to approximate the true score function; furthermore, $q(r,T)$ is approximated with the noise distribution $\epsilon{\sim}\mathcal{N}(0,I)$. Finally, a decoder d_ψ is used to map the latent variables back into the input space.

Conditional generation. Our model accommodates conditional generation: to do so, we capitalize on the multi-time masked diffusion model introduced in [12]. Specifically, the model leverages masked forward and backward diffusion processes to produce samples from the conditional distribution $q_\phi(z^M \,|\, z^C)$, being C and M the sets of conditioning and missing views to be generated, and, hence, z^C and z^M the respective latent variables. We define the masked forward SDE as:

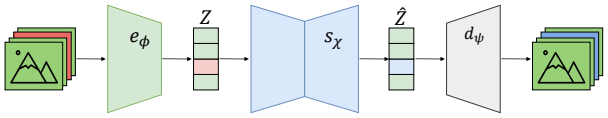

Fig. 1. Architecture of the employed latent diffusion model.

$$\begin{aligned}\mathrm{d}R_t = \mathcal{M}(M)\odot\left[\alpha(t)R_t\mathrm{d}t + g(t)\mathrm{d}W_t\right],\\ q(r,0) = q_\phi(r^M \,|\, z^C)\delta(r^C - z^C),\end{aligned} \tag{4}$$

where $R_0{=}\mathcal{C}(R_0^M, R_0^C)$, with $R_0^M{\sim}q_\phi(r^M \,|\, z^C)$, $R_0^C{=}z^C$, and $\mathcal{C}(\cdot)$ being the concatenation operator. Importantly, the mask $\mathcal{M}(M)$ is used to freeze or diffuse the latent variable z^C and z^M, respectively.

The reverse-time process of (4) is defined as follows:

$$\begin{aligned}\mathrm{d}R_t{=}\mathcal{M}(M)\odot\left[(-\alpha(T{-}t)R_t{+}\right.\\ \left.g^2(T{-}t)\nabla\log\big(q(R_t,T{-}t\,|\,z^C)\big)\right)\mathrm{d}t{+}g(T{-}t)\mathrm{d}W_t\right],\end{aligned} \tag{5}$$

with $R_0{=}\mathcal{C}(R_0^M, z^C)$ and $R_0^M{\sim}q(r^M, T \,|\, z^C)$. In this case, $q(r^M, T \,|\, z^C)$ is approximated by its corresponding steady-state distribution $\epsilon{\sim}\mathcal{N}(0,I)$, and the true conditional score function $\nabla\log\big(q(r,t\,|\,z^C)\big)$ is estimated with a conditional score network $s_\chi(r^M, t \,|\, z^C)$.

The diffusion model implements masked diffusion also using a multi *multi-time vector* $\tau{=}[t_1,\ldots,t_V]$, which concurrently indicates the diffusion time and which views are missing. Formally, the multi-time vector is defined as $\tau(M,t){=}t\left[\mathbb{1}(1{\in}M),\ldots,\mathbb{1}(V{\in}M)\right]$.

III. Methodology

In this section, we explain in detail our methodology, which relies on the latent diffusion model, built on the method introduced in [12], to generate synthetic views that are used to improve the performance of a classification model.

A. Latent diffusion model

At inference time, the architecture of the latent diffusion model is depicted in Fig. 1. We consider a set of $V{=}4$ views in total and, for instance, the set of missing views with $M{=}\{3\}$. The deterministic encoder e_ϕ encodes each conditioning reference view X^c, with $C{=}\{1,2,4\}$, producing their latent representations $Z^c{=}e_\phi(X^c)$; on the other hand, the missing view latent variable is represented using random noise, sampling from the Normal distribution $\mathcal{N}(0,I)$. The obtained latent representations are concatenated in order to obtain the variable Z, which is fed into the score-based diffusion model s_χ. Finally, the diffusion model generates the latent variable $\hat{Z}^m$ of the missing view X^m, and

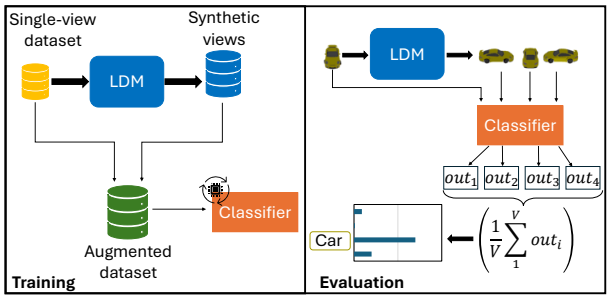

Fig. 2. Scheme of DIMVIDA for training (left) and evaluation (right).

the deterministic decoder d_ψ maps it back into the input space, producing the generated image $\hat{X}^m{=}d_\psi(\hat{Z}^m)$.

Training procedure. First, the encoder e_ϕ and the decoder d_ψ are jointly trained by optimizing the loss function in (1). Then, they are frozen, and we train the score-based diffusion model, which learns the conditional distribution of the missing latent variables $Z^M{=}\{Z^m\}_{m\in M}{\sim}p(z^M|z^C)$. To do so, we employ the multi-time masked diffusion approach introduced in [12], by using a complete training set and randomly setting some views as missing during the fine-tuning. Specifically, with probability $d{=}0.2$ we set $C{=}\emptyset$, i.e., there is no conditioning view; hence, we diffuse all latent variables; on the other hand, with probability $1{-}d$, we perform masked diffusion: first, we uniformly sample the set of conditioning views over all the possible sets; then, the remaining views, which are assumed to be missing, are diffused, while the latent variables of the conditioning views are frozen.

Formally, we train the score-based diffusion model s_χ by minimizing the following loss:

$$\mathcal{L} = \lambda(M,C) \cdot \left\| \mathcal{M}(M) \odot \left[\nabla \log\big(q(R_t \mid z^C)\big) - s_\chi(R_t, \tau(M,t)) \right] \right\|_2^2, \tag{6}$$

where R_t is obtained by first sampling from the distribution $q(r \mid Z,t)$ and aggregating it with the input Z using the mask $\mathcal{M}(M)$. Importantly, we use a scaling factor $\lambda(M,C){=}1{+}\frac{|C|}{|M|}$ to take into account the randomization of M and C that leads to the diffusion of different portions of the latent space.

B. Classification pipeline

Once the latent diffusion model is trained, it is endowed with the capacity to generate missing views given the observed ones. Thus, we exploit this conditional generation ability to augment the classification pipeline.

The training procedure scheme of DIMVIDA is presented in Fig. 2(left). Given a single-view dataset, that is, composed of N samples with one view per sample, we use the LDM to generate for each sample 3 other images from different viewpoints. Then, we train a classification model using such an augmented training set. As shown in Fig. 2(right), during the evaluation step, we also augment the test set, following the same procedure used for the training set; thus, for each sample in the test data we produce a set of 4 views, including the real image and 3 synthetic ones. Next, for each set of multi-view images, we evaluate the model individually for every view and we compute the mean of the 4 classification model outputs, which is finally used to predict the class of the sample.

IV. Experiments

In this section, we first describe the dataset used for the experiments and we evaluate the generation performance of the diffusion model. Finally, we assess the effectiveness of our augmented classification pipeline.

Datasets. We assess DIMVIDA performance using the Neural 3D Mesh Renderer Dataset (NMR) [20]. NMR consists of objects of the 13 largest classes of ShapeNet [21] dataset, a collection of 3D objects, from which 64x64 2D images are rendered at 24 fixed views. For our experiments, we only use 4 views, i.e., front, back, right and left views.

Specifically, the dataset is split into three partitions:

- NMR_d, it includes for each sample all 4 views and is used to train both the autoencoder and the diffusion model;
- NMR_c, it includes only one random view per sample and is used to train the classifiers;
- NMR_e, it includes only one random view per sample and is used to evaluate the classifiers.

A. Training details of latent diffusion model

For the autoencoder, we use the same architecture used in [12] for the CUB [22] dataset and train the model using the Laplace distribution to estimate the likelihood. We used *TrivialAugmentWide* from the Torchvision library for data augmentation. We set the dimensionality of the latent space to 64; we perform 1000 training epochs, with learning rate $1\text{e}{-}4$ and batch size equal to 64. Regarding the score-based diffusion model, we borrow the architecture from [12] used for the CUB dataset. However, we extend the input dimension from 1024 to 1536. During training, we perform 1000 epochs, with the learning rate and batch size, respectively, equal to $1\text{e}{-}4$ and 64.

B. Results

Conditional image generation performance. First, we evaluate the performance of the LDM by assessing the quality of the conditionally generated images by computing the Fréchet Inception Distance (FID) [23], Peak Signal-to-Noise Ratio (PSNR), the Structural Similarity Index Measure (SSIM) [24] and the Learned

TABLE I
PERFORMANCE OF LATENT DIFFUSION MODEL

Metric	Available images		
	1	2	3
FID (↓)	16.43	15.14	14.51
LPIPS (↓)	0.074	0.056	0.048
SSIM (↑)	0.832	0.884	0.912
PSNR (↑)	24.028	26.776	28.491
Coherence (%↑)	85.49	88.33	89.14

Perceptual Image Patch Similarity (LPIPS) [25]. The FID score evaluates both the quality and diversity of the generated data; PSNR is a widely used metric based on the pixel-wise difference between two images; SSIM measures the similarity of two images by comparing luminance, contrast, and structure. SSIM aims to better reflect human visual perception, which is also the goal of LPIPS. However, the latter computes the difference between the features obtained from a layer of a pre-trained image convolutional neural network, namely SqueezeNet [26] in our implementation.

Furthermore, we verify that such generated data preserve the semantics of the conditioning images, i.e., the observed images: to do so, we measure the so-called coherence. Specifically, we use a pre-trained classifier Γ fine-tuned on our dataset to check that the generated images are classified coherently with the conditioning images. Formally, for N generated images, the coherence is computed as follows:

$$\textit{coherence}(\hat{X}^m|X^C) = \frac{1}{N}\sum_{1}^{N} \mathbb{1}_{\{\Gamma(\hat{X}^m)=y_{X^C}\}}, \quad (7)$$

where $\hat{X}^m$ is the image generated by the diffusion model conditionally with respect to set X^C, while y_{X^C} is the true label, i.e., the class of the conditioning set.

The computed metrics are reported in Tab. I, which shows that the performance improves when more images are observed, highlighting that the model efficiently aggregates information from different views. In general, the reported metrics also indicate the capability of the latent diffusion model to generate high-quality data, while maintaining the semantics of the conditioning images when generating the missing views.

Classification performance To evaluate the impact of synthetic images on the augmented classification pipeline, we first consider a classifier Ω_s, trained on the single-view NMR$_c$ dataset, and a classifier Ω_m, trained on the augmented dataset NMR$_{ca}$, which includes both all NMR images$_c$ and the synthetic images obtained with the diffusion model conditioned on NMR$_c$ data. During classifier training, we use the Adam optimizer with learning rate $1\mathrm{e}{-5}$ and batch size set to 128,

As for the evaluation protocol, we consider two approaches. The first is the standard procedure, where trained models are tested on the NMR$_e$ single-view evaluation data set. The second method uses the latent diffusion model to augment the NMR$_e$ dataset, as done with NMR$_c$, obtaining the dataset NMR$_{ea}$.

TABLE II
CLASSIFIERS ACCURACY (%) - TRAINING FROM SCRATCH

Architecture	Model - training set	Test set	
		NMR$_e$	NMR$_{ea}$
ResNet18	Ω_s - NMR$_c$	83.89	87.21
	Ω_m - NMR$_{ca}$	85.18	**88.03**
MobileNetV3	Ω_s - NMR$_c$	71.01	79.66
	Ω_m - NMR$_{ca}$	77.95	**85.45**

TABLE III
CLASSIFIERS ACCURACY (%) - FINE-TUNING

Architecture	Model - training set	Test set	
		NMR$_e$	NMR$_{ea}$
ResNet18	Ω_s - NMR$_c$	89.98	89.89
	Ω_m - NMR$_{ca}$	90.83	**90.96**
MobileNetV3	Ω_s - NMR$_c$	87.14	88.15
	Ω_m - NMR$_{ca}$	88.08	**90.00**

We perform experiments with two state-of-the-art classification models, namely ResNet18 and MobileNetV3, training them from scratch. The results reported in Tab. II underline that the use of augmented multiview datasets both for training and evaluation leads to the best classification performance. Moreover, we also find that performing the augmentation only during the evaluation improves the performance more than augmenting solely the training set.

Finally, we also fine-tune the models starting from the pre-trained weights provided by the Torchvision library. The obtained results are shown in Tab. III, which further demonstrates the benefits of our augmented classification pipeline, even if the gain is lower than the case of the training from scratch: very likely, the initialization of the model with the pre-trained weights equips the classification models with an already appropriate visual understanding ability, limiting further enhancements.

V. RELATED WORK

Generative models are a class of machine learning algorithms designed to learn the underlying distribution of the input data, which enables the generation of new data samples that resemble the training data.

Over the years, Generative Adversarial Networks (GANs) [6] and Variational AutoEncoders (VAEs) [7] have dominated the field; however, recently, diffusion models [1], [27] have emerged as a superior alternative. [28] introduces the concept of the "generative learning

trilemma", which highlights the three main challenges in generative modeling, namely high-quality sample generation, sample diversity, and fast generation. While GAN-based methods are characterized by poor mode coverage, i.e., low diversity of generated samples, and VAEs tend to produce samples of lower quality, score-based diffusion models are capable of generating both high-quality and diverse images, though they are slower in the generation process.

Augmentation with synthetic data. Many works have implemented data augmentation techniques that rely on GANs for classification tasks, such as [8], [9]. [29] trains a classifier using only synthetic data produced with a GAN, resulting in improved performance compared to the original dataset. [19] proposes a method similar to ours, generating an augmented dataset by producing images of real data from novel viewpoints; however, data augmentation is performed by means of a GAN and, more importantly, only involves the training set. Our approach, instead, takes advantage of the multi-view synthetic data also during evaluation.

Recently, diffusion models have replaced GANs for data augmentation approaches based on synthetic images, thanks to their superior generation capability, both in terms of data quality and diversity. For instance, [10] demonstrates the benefits of using synthetic images generated by a state-of-the-art text-to-image diffusion model for image classification. In particular, the impact of synthetic data is analyzed in three scenarios, namely zero-shot learning, few-shot learning and large-scale model pre-training for transfer learning. [11] leverages diffusion models to perform data augmentation by changing the data semantics, increasing data diversity.

VI. Conclusions

In this paper, we have presented DiMViDA, a novel data augmentation technique based on a diffusion-based novel view synthesizer method. Our approach is specifically designed for a classification task: specifically, we envision augmenting data both during training and inference, as the evaluation step takes into account the single real test image and the ones generated by the diffusion model, which represent the same object from different viewpoints. Importantly, the experiments performed have supported the effectiveness of the proposed approach, with gains in classification accuracy up to 20%.

As for future work, we will extend our experimental campaign to further corroborate the benefits of our augmented classification pipeline. In particular, we will use additional datasets and state-of-the-art diffusion models. Also, we will investigate how the diversity of the generated images impacts the classification performance.

Acknowledgements

This work was supported by the European Commission under Grant Agreement No. 101095363 (ADROIT6G project).

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[2] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," *Advances in neural information processing systems*, vol. 34, pp. 1415–1428, 2021.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[4] G. Franzese, S. Rossi, L. Yang, A. Finamore, D. Rossi, M. Filippone, and P. Michiardi, "How much is enough? a study on diffusion times in score-based generative models," *Entropy*, vol. 25, no. 4, p. 633, 2023.

[5] G. Franzese, G. Corallo, S. Rossi, M. Heinonen, M. Filippone, and P. Michiardi, "Continuous-time functional diffusion processes," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=VPrir0p5b6

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[8] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *arXiv preprint arXiv:1711.04340*, 2017.

[9] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid, "A bayesian data augmentation approach for learning deep models," *Advances in neural information processing systems*, vol. 30, 2017.

[10] R. He, S. Sun, X. Yu, C. Xue, W. Zhang, P. Torr, S. Bai, and X. QI, "IS SYNTHETIC DATA FROM GENERATIVE MODELS READY FOR IMAGE RECOGNITION?" in *The Eleventh International Conference on Learning Representations*, 2023.

[11] B. Trabucco, K. Doherty, M. A. Gurinas, and R. Salakhutdinov, "Effective data augmentation with diffusion models," in *The Twelfth International Conference on Learning Representations*, 2024.

[12] M. Bounoua, G. Franzese, and P. Michiardi, "Multi-modal latent diffusion," *Entropy*, vol. 26, no. 4, p. 320, 2024.

[13] Z. Tang, Z. Yang, C. Zhu, M. Zeng, and M. Bansal, "Any-to-any generation via composable diffusion," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[14] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9298–9309.

[15] X. Kong, S. Liu, X. Lyu, M. Taher, X. Qi, and A. J. Davison, "Eschernet: A generative model for scalable view synthesis," *arXiv preprint arXiv:2402.03908*, 2024.

[16] G. Di Giacomo, G. Franzese, T. Cerquitelli, C. F. Chiasserini, P. Michiardi *et al.*, "Dimvis: Diffusion-based multi-view synthesis," in *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling 2nd SPIGM@ ICML*. ACM, 2024.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[19] W. Xiong, Y. He, Y. Zhang, W. Luo, L. Ma, and J. Luo, "Fine-grained image-to-image transformation towards visual recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5840–5849.

[20] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3907–3916.

[21] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[22] Y. Shi, B. Paige, P. Torr *et al.*, "Variational mixture-of-experts autoencoders for multi-modal deep generative models," *Advances in neural information processing systems*, vol. 32, 2019.

[23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[25] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[26] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[27] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[28] Z. Xiao, K. Kreis, and A. Vahdat, "Tackling the generative learning trilemma with denoising diffusion GANs," in *International Conference on Learning Representations*, 2022.

[29] F. H. K. D. S. Tanaka and C. Aranha, "Data augmentation using gans," *arXiv preprint arXiv:1904.09135*, 2019.