Synthetic Image Detection using Real and Fourier Domain Features

Shahriar Kabir Nahin[1,*], Sanjay Acharjee[2], Sawradip Saha[3,†], Fazle Rabbi[1], Asif Quadir[1], Indrojit Sarkar[1], Mohammad Ariful Haque[1,‡]

[1]*Department of Electrical and Electronic Engineering,* [2]*Department of Civil Engineering,* [3]*Department of Mechanical Engineering*
Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh

[*]shahriarkabir0.sk.sk@gmail.com, [†]sawradip0@gmail.com, [‡]arifulhoque@eee.buet.ac.bd

***Abstract*—The ability to classify Synthetic images from a pool of Synthetic and Real images is extremely important for reasons ranging from recognizing harmless photo-editing to forensic investigation. Other than manually editing images, the most popular way to generate synthetic or partially manipulated images are through deep-learning. Generative Adversarial Networks, Diffusion Methods and Gated Convolution are mostly used to create fake images. So the task of Synthetic and partially manipulated image detection should begin by studying and experimenting with images generated by this method. In this work, a deep learning based binary classification pipeline has been proposed for classification of real and Synthetic or partially manipulated images have been proposed. In our approach, features extracted from both from the image and its Fourier transformed version along with various augmentation such as cut-mix and flipping has been used. These features has been extracted by popular feature extractor such as EfficientnetB7 and Mobilentv3 followed by concatenation and series linear layers to finally obtain the prediction.**

***Index Terms*—Synthetic Image, GAN, Camera Footprint, FFT, Entropy, DNN.**

I. Introduction

In the past couple of years, there has been an extensive study on the development of synthetic image generation and many deep learning based algorithms have been developed to this aim. Deep learning techniques such as Generative Adversarial Networks (GAN) [1], Vision Transformers [2], Diffusion Models [3] have advanced fake image synthesis processes. Some of the algorithms synthesize fake images from scratch, while others change the properties of existing pictures. Nowadays, several fascinating applications of these technologies exist. However, this technology may also be employed for sinister reasons, such as the creation of fraudulent social media profiles and fake news. Synthetic images can now deceive even the most vigilant observer, let alone the average Internet user. These have sparked a great deal of public anxiety as it is quite impossible to distinguish between real and fraudulent pictures. Therefore, there is an immediate demand for automated technologies that can discriminate between true and modified materials with reliability. [4] In fact, despite their excellent image quality, synthetic photos include unique evidence left by the production process that may be used to recognize them. Occasionally, they exhibit obvious abnormalities, such as color anomalies or asymmetry. Nevertheless, due to the rapid evolution of technology, these visible flaws will certainly disappear in the near future.

II. Related Work

The task of detecting synthetic or partially manipulated images has been addressed before in literatures related to digital forensics [5] as well as deepfake detection [6]. Two popular approaches used in these investigations were discovering various limitation of digitally generated images as well as extracting useful features from fourier domain.

The fact that GANs produce a limited range of intensity values and do not generate saturated or underexposed regions is exploited in [7] by measuring the frequency of saturated and underexposed regions in both real and synthetic images. A large percentage of natural face images contain pixels with extreme values, and their absence signifies artificiality. Existing GANs can't effectively preserve the inherent correlation between color bands. This property is leveraged in [8], where the image's chrominance components are high-pass filtered and their co-occurrence matrices are constructed to extract discriminative information. Because invisible artifacts are frequently present in high-frequency signal components [9], co-occurrences of high-pass filtered images are common in image forensics. As a result, co-occurrence matrices derived from RGB channels are used in [10] as the input to a CNN, and in [11], co-occurrences across color bands are computed to capture discriminative information.

A frequency-domain analysis [12] is conducted to check for artifacts in various network designs. A detector is proposed in [13] that feeds a CNN with the frequency spectrum rather than image pixels is. Training in a CNN-based classifier with Fourier spectra from both actual and fake images using an adversarial autoencoder. Also, [14] demonstrates that the spectral distributions of real images are not accurately mimicked by GAN images. In [15], the Fourier spectrum's decay function is fitted using a parametric model, and a classifier is trained using the fitting parameters.

The aforementioned methods actually perform poorly with GAN images originating from architectures not featured in our trainset. An autoencoder-based architecture [16] [17] was

proposed to be generalized for all architectures. In [18], an alternative method, incremental learning, is utilized. But both of them require samples of the new GAN architecture. L. Chai et el. [19] propose a fully convolutional patch-based classifier. The authors demonstrate that performance can be enhanced by concentrating on local patches as opposed to global structures. In [20], the authors proposed a different approach. As all the GAN architectures are CNN-based, they trained a classifier to detect CNN fingerprints. This model acquires the ability to generalize to unexplored datasets, architectures, and tasks via rigorous pre- and post-processing, data augmentation, and training data diversification. But efficiency degrades on social networks as images are continually shrunk and resized.

In [21], the authors modified the model presented in [20], incorporating a number of the essential components of the most promising approaches. They eliminated Imagenet pre-trains, incorporated an initial layer for residual extraction [22], omitted downsampling in the first layer [23] and improved augmentation (Gaussian noise adding, geometric transformations, brightness and contrast changes). In addition, they replaced ResNet50 with Xception (Xception no-down) and Efficient-B4 (Efficient no down) in the backbone network.

III. Dataset

In our experimental analysis, we utilize real image datasets and synthetic image datasets. We use several well-known public datasets as the basis for our real-world training images named "COCO" [24], "ImageNet" [25] [26], "FFHQ" [27], "LSUN" [28]. For synthetic datasets, some image datasets of 'StyleGAN2' [29], 'StyleGAN3' [30], 'Gated Convolution' [31] [32], 'GLIDE' [33], and 'Taming Transformers' [34] are provided by the competition authorities. Table I illustrates an accurate description of the synthetic data used in the experiment.

Recent research has shown that diffusion models may create high-quality synthetic pictures, particularly when coupled with a guiding strategy that trades variety for fidelity. As a result, the Glide image is difficult to distinguish from the real one. Typically, gated convolution is used to eliminate distracting elements, adjust picture layouts, exclude watermarks, and modify faces, all of which are performed on actual images. As a result, because it leaves so few traces, distinguishing it from the authentic version is difficult. Using a transformer, it is possible to generate high-resolution images.

TABLE I
Datasets of GAN Generated Image

GAN Model	Subfolder	Size	Classes	# Images
Gated Conv		256×256	Random	2000
Glide	Impainting	256×256	Random	1000
	Text2Image	256×256	Random	1000
StyleGAN2	Dlr0	256×256	Horse	10116
	Dlr1	320×512	Cars	10063
	Dlr2	256×256	Church	10024
	Dlr3	1024×1024	Face	10263
	Dlr4	256×256	Cats	10035
StyleGAN3	Dlr0	512×512	Cats,Dogs	5100
	Dlr1	512×512	Cats,Dogs	5127
	Dlr2	1024×1024	Painted Face	5100
	Dlr3	1024×1024	Painted Face	5100
	Dlr4	1024×1024	Painted Face	5100
	Dlr5	1024×1024	Face	5148
	Dlr6	1024×1024	Face	5162
	Dlr7	1024×1024	Face	5224
	Dlr8	1024×1024	Face	5333
	Dlr9	1024×1024	Painted Face	5100
Taming	Imagenet	256×256	1000 class	50000
	ffhq	256×256	Face	50000

IV. Methodology

A. Data Preparation

The test images of the competition are compressed to JPEG, cropped certain portion of the image and resized to 200×200. So, before training we have prepared the images accordingly.

B. Data Augmentation

As this is a task of synthetic image detection we have to be sincere about selecting augmentations so that important features of the image don't get destroyed. Here, horizontal and vertical flip are used during training. Also, cut-mix is used as an augmentation where we cut an image into pieces and rebuild the image by shuffling the order of the pieces. Figure 1 shows the augmentations.

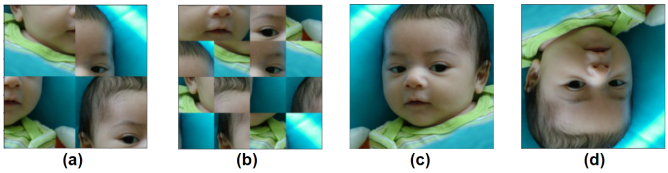

Fig. 1. Different types of augmentation. (a,b: Cut-mix, c,d: horizontal and vertical filp)

C. Features Extraction

As the images are resized and also compressed to JPEG, many important features are already vanished from the images. Such as camera footprint. So, we decide to get information as much as possible from an image. We have extracted some Real and Fourier domain features from the images.

An image captured from a camera always contains a footprint. Because, when we take an image light passes through the camera lens. Then it passes through a filter which contains small grids of RGB filters periodically. Without the filter we cannot get an exact colourful image. After passing the filter light comes to a Sensor and then after some Signal processing steps and interpolation we can get a colourful image (See Figure: 2). As GAN generated images cannot follow the basic procedures to generate an image similar to real image, there will be significant difference in RGB channels between a real image and GAN generated image. Also, there will be some

difference in pixel distribution and transition of pixel values in a GAN Generated image. GAN generated image will not have much details and different edges will not be generated properly.

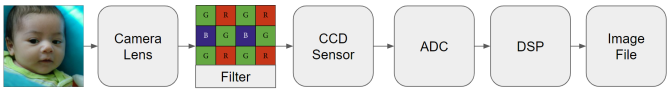

Fig. 2. Generation process of camera captured image.

1) Real Domain Features: To extract the sharp transitions of an image we first get the Fourier transform of a channel. Then we make the lower frequencies zero forcefully by using a mask. Then we apply Inverse Fourier Transform to that. Thus, we get an image which contains only some edges of the main image (See Figure: 3). We do it for every channel and different mask size. In our training we have used a total of 10 masks with window size 2×2, 4×4, 6×6 etc.

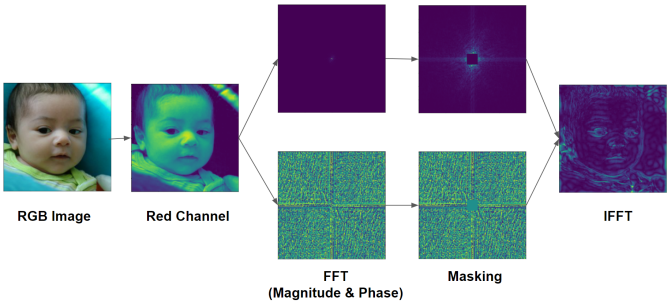

Fig. 3. Feature extraction of a single channel by suppressing lower frequencies.

After extracting the edges we calculate the entropy of every image by taking the image in gray scale (See Figure: 4). Entropy shows complexity or randomness of different parts in an image. We can write it in a mathematical formula 1.

$$E = -\sum_{i=0}^{255} p_i \log p_i \tag{1}$$

where p is the probability of occurring any pixel value and E is the entropy of the whole image. We can calculate the entropy in parts using small windows. In our training we have used three types of disks or windows (1,2,3) to calculate the entropy in different portions. More disks can be taken but that isn't feasible as it takes much time.

Then we concatenate the main RGB image, Masked IFFT of three channels for different windows and Entropy for different disk size to get our Real Domain Features. Then we normalize each feature by dividing the maximum values.

2) Fourier Domain Features: We already know camera uses filters of periodic grid of RGB colour to generate a coloured image. That's why we decide to feed some Fourier Domain Features in our model. In this case we simply calculate the magnitude and phase of each channel's Fourier Fransform of a RGB image and concatenate them. These, features are also normalised between 0 to 1.

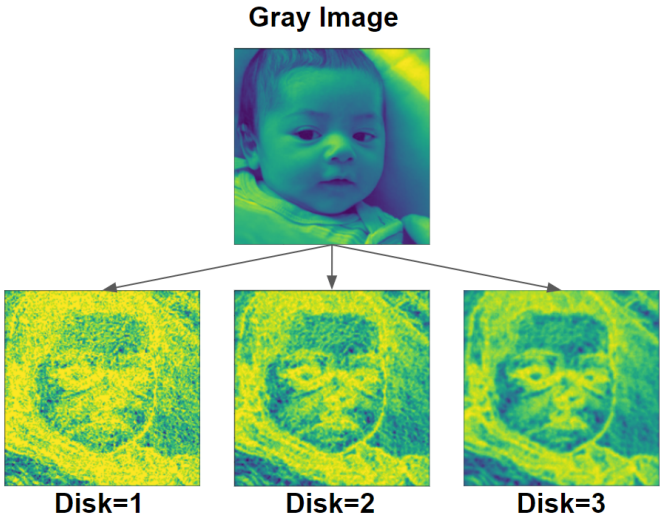

Fig. 4. Entropy of an image.

D. Model Architecture

We are using features from two different domain. So, we decide to use two separate backbones for these two domains (See Figure: 5). For Real Domain features we have used an Efficientnet-b7 [35] backbone which is our main backbone. As we have a very large Dataset, we decide to use a large backbone here. Beside this one we have used a Mobilenet-v3-large [36] backbone for Fourier Domain Features. We have tried Efficientnet-v2-large, ConvNext [37] instead of Efficientnet-b7 and other small backbones of Efficientnet, CovNext-small instead of Mobilenet, but those combinations cannot perform well.

For both the model the first layer is down-sampling layer with a stride of 2×2. we have replaced the first layer with a layer of stride 1×1 by keeping kernel size same. By feeding our data to this backbones we get two separate feature vectors. After that, we concatenate these two feature vectors and then by feeding the feature vector into a dropout and linear layer we get our final prediction.

E. Loss and Metric

To calculate our loss we have used Binary Cross Entropy Loss and the accuracy is average correct prediction.

V. Experiments and Results

A. Training and Evaluation

We have trained our model for a total of 100 epochs. In every epoch we use 10 thousand images for training and 4 4 thousand images for validation. We can see that we don't have good number of images of every GAN Models. So, we have ensured data balancing during our training process so that during training phase, images come from all parts of the dataset with equal probability. We have used 75 percent of our whole data for training and 25 percent for validation. In every epoch 50 percent data comes from Camera captured Images

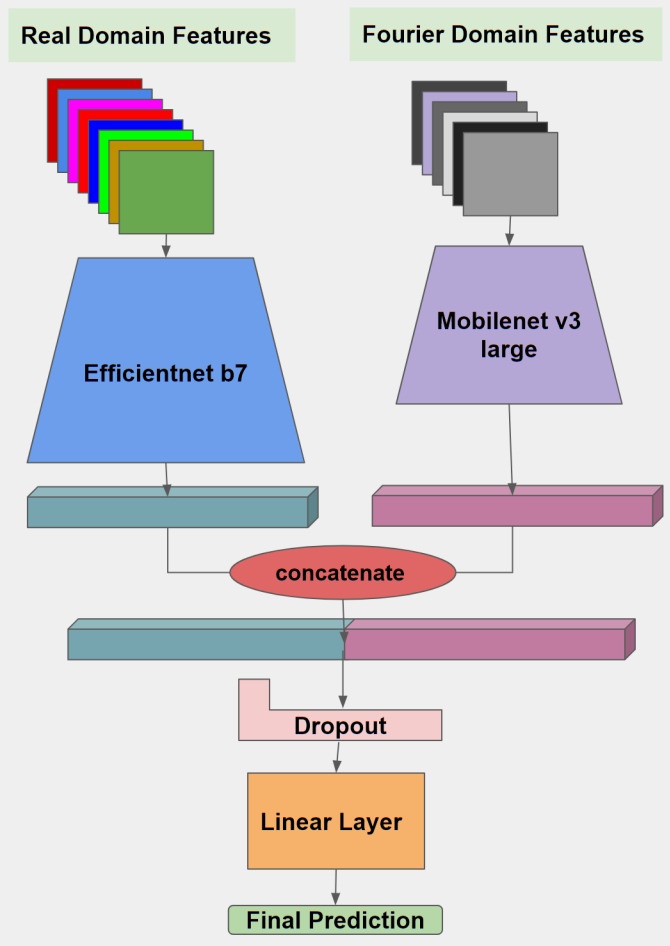

Fig. 5. Model Architecture.

and 50 percent comes from GAN Generated Images. Figure 6 shows the training and validation loss curve.

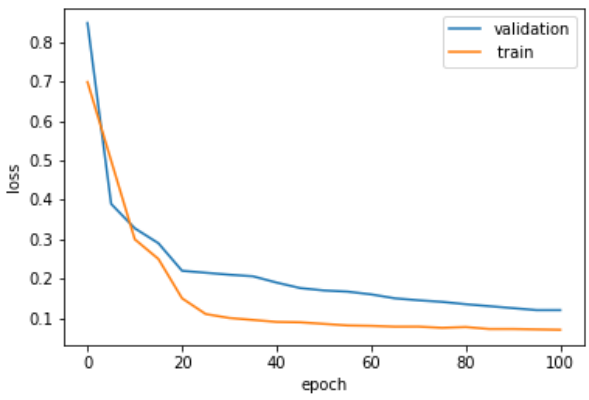

Fig. 6. Training and Validation Loss

B. Results

Table II shows our final accuracy for training and validation for different part of dataset. We have got a validation accuracy of 96.46 percent and Training accuracy of 96.91 percent in our Dataset. Here, the model performs well in detecting camera generated images III. Also, we can see in case of GAN Generated images our model do well in StyleGAN2, StyleGAN3 and Taming. But in Glide and Gated Convolution the performance is 6 to 10 percent lower than the others.

TABLE II
ACCURACY IN FINAL TRAINING AND VALIDATION

Data	Accuracy
Training Data	96.91
Validation Data	95.48

The results are average accuracy of randomly sampled data from Datasets multiple times.

TABLE III
VALIDATION ACCURACY IN DIFFERENT DATASETS.

Data	Accuracy
Real Images	95.52
Gated Conv	90.42
Glide	92.81
StyleGAN2	98.41
StyleGAN3	96.24
Taming	99.48

The results are average accuracy of randomly sampled data from validation set multiple times.

VI. Conclusion

Synthetic Images detection is never an easy task. New and stronger GAN Models are coming every year and the challenge is becoming tougher. Although, without JPEG compression and resizing image the detection task is easier but that is not practical. In real world we will have different images of lower resolution which are manipulated in multiple ways. As a result, we will not have much camera footprint to detect an image easily. So, if we can develop the models for this type of challenging condition that will help more. In case of our model, it doesn't perform very good in the datasets of Glide and Gated convolution. We are assuming this confusion arises because, in Gated Convolution and Glide all portions of a image aren't fully GAN generated every time. We will work to extract more features and develop our model which can solve this problem.

VII. Acknowledgment

We want to thank the organizers of VIP Cup 2022 as well as IEEE for launching a competition with a problem which is really important in this new era.

References

[1] 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.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[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 (CVPR)*, June 2022, pp. 10 684–10 695.

[4] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva, "Are gan generated images easy to detect? a critical analysis of the state-of-the-art," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[5] H. Farid, "Image forgery detection," *IEEE Signal processing magazine*, vol. 26, no. 2, pp. 16–25, 2009.

[6] S. Lyu, "Deepfake detection: Current challenges and next steps," in *2020 IEEE international conference on multimedia & expo workshops (ICMEW)*. IEEE, 2020, pp. 1–6.

[7] S. McCloskey and M. Albright, "Detecting gan-generated imagery using saturation cues," in *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019, pp. 4584–4588.

[8] H. Li, B. Li, S. Tan, and J. Huang, "Detection of deep network generated images using disparities in color components, 2018," pp. 1–13, 1808.

[9] D. Cozzolino, G. Poggi, and L. Verdoliva, "Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection," in *Proceedings of the 5th ACM workshop on information hiding and multimedia security*, 2017, pp. 159–164.

[10] L. Nataraj, T. M. Mohammed, B. Manjunath, S. Chandrasekaran, A. Flenner, J. H. Bappy, and A. K. Roy-Chowdhury, "Detecting gan generated fake images using co-occurrence matrices," *Electronic Imaging*, vol. 2019, no. 5, pp. 532–1, 2019.

[11] M. Barni, K. Kallas, E. Nowroozi, and B. Tondi, "Cnn detection of gan-generated face images based on cross-band co-occurrences analysis," in *2020 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2020, pp. 1–6.

[12] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258.

[13] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in gan fake images," in *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2019, pp. 1–6.

[14] T. Dzanic, K. Shah, and F. Witherden, "Fourier spectrum discrepancies in deep network generated images," *Advances in neural information processing systems*, vol. 33, pp. 3022–3032, 2020.

[15] R. Durall, M. Keuper, and J. Keuper, "Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7890–7899.

[16] D. Cozzolino, J. Thies, A. Rössler, C. Riess, M. Nießner, and L. Verdoliva, "Forensictransfer: Weakly-supervised domain adaptation for forgery detection," *arXiv preprint arXiv:1812.02510*, 2018.

[17] M. Du, S. Pentyala, Y. Li, and X. Hu, "Towards generalizable forgery detection with locality-aware autoencoder," 2019.

[18] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental learning for the detection and classification of gan-generated images," in *2019 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2019, pp. 1–6.

[19] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," in *European conference on computer vision*. Springer, 2020, pp. 103–120.

[20] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8695–8704.

[21] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva, "Are gan generated images easy to detect? a critical analysis of the state-of-the-art," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[22] L. Verdoliva, "Media forensics and deepfakes: an overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[23] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[27] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[28] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[29] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[30] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 852–863, 2021.

[31] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[32] ——, "Free-form image inpainting with gated convolution," *arXiv preprint arXiv:1806.03589*, 2018.

[33] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," *arXiv preprint arXiv:2112.10741*, 2021.

[34] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[35] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[36] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[37] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022. [Online]. Available: https://arxiv.org/abs/2201.03545