In-Depth Review and Application of Generative Adversarial Networks (GANs) for Image Generation CSE 575 Team 31

Akshat Nambiar
ASU ID: 1225484000

Ryan Collins
ASU ID: 1225687957

Sahithya Cherukuri
ASU ID: 1226054065

Sai Prakash Ravichandran
ASU ID: 1225761147

1 Introduction

The goal of our project is to examine how Generative Adversarial Networks (GANs) may be used to produce synthetic data of excellent quality and near resemblance to actual data. We specifically want to build a GAN architecture that can produce images or designs for cars. We want to generate realistic automotive pictures that may be used in numerous applications, such as virtual reality, gaming, and design, by training a generator and a discriminator network.

The importance of this project cannot be overstated. With the increasing demand for realistic, high-quality images in various industries, such as automotive design and virtual reality, the ability to generate such images quickly and accurately is becoming a critical requirement. The GAN architecture offers a special approach that can rapidly and accurately create a wide variety of pictures in response to the growing demand for such graphics. For the development of generative deep learning and artificial intelligence, this study is crucial.

Potential applications also have a huge amount of promise. Once we have produced high-quality car images, the technology can be expanded upon and applied to other industries like fashion, medicine, and the arts. The success of this initiative will thus have broad ramifications and provide new prospects for other businesses to use AI-generated pictures in their work.

Our research proposal highlights the value of GANs in producing realistic, high-quality photographs in its conclusion. We want to show off this technology's capabilities in the context of car design since it has such enormous potential. By doing this, we intend to advance generative deep learning capabilities and advance the development of AI generally.

2 Objectives

- To review the literature on GAN architecture and its various versions.
- To identify a suitable dataset of automobile photos for training the GAN model.
- To develop a GAN model that can generate realistic car images.
- To evaluate the performance of the GAN model using appropriate metrics.
- To compare the results of the GAN model with other existing image generation methods.
- To identify potential applications of GAN-generated car images.
- To suggest possible improvements for the GAN model and future research directions.

3 Literature Review

1. Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

"Generative Adversarial Networks" by Ian Goodfellow, et al. is a seminal paper that introduced the concept of GANs and how they work. The paper describes the two key components of a GAN: the generator and the discriminator networks. The generator network learns to generate data that is similar to the training data, while the discriminator network learns to distinguish between real and fake data.

The paper also introduces the concept of adversarial training, in which the generator and discriminator networks are trained in a game-like setting, where the generator tries to generate data that can fool the discriminator, and the discriminator tries to correctly classify the generated data as fake. The authors propose a training algorithm that alternates between updating the generator and discriminator networks, with the goal of finding a Nash equilibrium between the two networks. The paper also shows that GANs can be used for a variety of tasks, including image synthesis, style transfer, and image inpainting.

Overall, "Generative Adversarial Networks" is an important paper that introduced a powerful new approach to generative modeling. The concept of GANs has since been extended and refined in numerous subsequent papers, leading to significant advances in areas such as image synthesis, text generation, and music generation.

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

The paper builds on the work of the original GAN paper by Goodfellow et al. and proposes modifications to the architecture of the generator and discriminator networks to improve the quality of generated images.

The paper introduces a novel approach to training GANs that involves replacing the traditional pooling layers in the discriminator network with convolutional layers, which allows for better feature representation and avoids down-sampling artifacts. The authors also propose using batch normalization and rectified linear units (ReLU) activation functions to improve the stability of the training process.

The paper demonstrates that the proposed architecture is effective at generating high-quality images of cars from random noise. The authors evaluate their model using standard metrics such as the Inception Score and Frechet Inception Distance and show that it outperforms previous GAN models on these metrics.

3. Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive growing of GANs for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

"Progressive Growing of GANs for Improved Quality, Stability, and Variation" by Tero Karras, et al. is a paper that proposes a technique for training GANs that involves starting with low-resolution images and progressively increasing the resolution as training progresses. The authors demonstrate that this approach results in higher-quality and more varied images compared to traditional GAN training methods.

The paper introduces a new architecture for the generator and discriminator networks that allows for seamless transitioning between different image resolutions during training. The authors also propose a new metric for evaluating the quality of generated images, called the Fréchet Inception Distance (FID), which measures the similarity between the distributions of real and generated images in feature space.

4. Park, Taesung, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. "Semantic image synthesis with spatially-adaptive normalization." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2337-2346. 2019.

The paper introduces a new normalization technique called Spatially-Adaptive Normalization (SPADE), which allows for the adjustment of feature normalization based on semantic information provided as input to the generator network. This approach allows for the generation of images with specific semantic attributes, such as different textures, colors, and shapes.

The paper demonstrates the effectiveness of the proposed method on a variety of image synthesis tasks, including face generation, animal generation, and scene generation. The authors show that the method can be used to generate images with high-quality and diverse semantic content.

The paper also explores the use of a perceptual loss function, which measures the similarity between the generated image and a target image in terms of perceptual features extracted from a pre-trained neural network. The authors show that the use of a perceptual loss function improves the quality of generated images and helps to avoid mode collapse.

5. Zhang, Han, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks." In Proceedings of the IEEE International Conference on Computer Vision, pp. 5907-5915. 2017.

The authors propose a novel architecture that uses two separate GANs to generate high-resolution images from textual descriptions. The first GAN generates low-resolution images that serve as a conditioning input to the second GAN, which generates high-resolution images. The authors also introduce a novel conditioning technique called "text-embedding-based conditioning" that allows for the conditioning of the generator network using textual descriptions.

The paper demonstrates the effectiveness of the proposed model on the task of generating photo-realistic images of birds and flowers from textual descriptions. The authors show that the model is capable of generating high-quality images that are visually similar to the objects described in the text.

Overall, this is an important paper that demonstrates the potential of GANs for image synthesis beyond just generating images from noise. The paper's proposed architecture and conditioning technique have since been used in numerous follow-up studies, and the paper's results have led to significant advancements in the field of text-to-image synthesis.

6. Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs trained by a two time-scale update rule converge to a local Nash equilibrium." In Advances in Neural Information Processing Systems, pp. 6626-6637. 2017.

The authors propose a two time-scale update rule (TTUR) that updates the generator and discriminator at different rates. The generator is updated less frequently than the discriminator, which helps stabilize the training process and avoid mode collapse. The authors demonstrate the effectiveness of the proposed algorithm on several benchmark datasets, showing that it leads to improved stability and convergence compared to other GAN training methods.

The paper also includes a thorough theoretical analysis of the proposed algorithm, demonstrating that it converges to a local Nash equilibrium under certain conditions. This analysis provides a deeper understanding of the dynamics of GAN training and the role that the TTUR algorithm plays in achieving stable and effective convergence.

7. Wu, Jiajun, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling." In Advances in Neural Information Processing Systems, pp. 82-90. 2016.

The authors propose a 3D GAN model that learns a probabilistic latent space of object shapes. The model is trained on a dataset of 3D object shapes and learns to generate new objects by sampling from the latent space. The authors demonstrate the effectiveness of the proposed model on several benchmark datasets, showing that it can generate high-quality 3D models of objects with a wide range of shapes and styles.

The paper also includes a detailed analysis of the learned latent space, showing that it captures meaningful geometric and structural properties of objects. This analysis provides insights into the nature of the latent space and how it can be used for tasks such as shape interpolation and manipulation.

8. Cuturi, Marco, Arnaud Doucet, and Sebastian Nowozin. "Deep generative models for distribution-preserving operations: Introducing wasserstein flow." arXiv preprint arXiv:1705.07079 (2017).

The authors propose a new approach called Wasserstein flow, which involves minimizing the Wasserstein distance between the model distribution and the true data distribution. The Wasserstein distance is a metric that measures the distance between two probability distributions, and has some desirable properties for GAN training, such as being differentiable almost everywhere.

The paper demonstrates the effectiveness of the proposed method on several benchmark datasets, showing that it can generate high-quality images with improved stability and diversity compared to traditional GAN training methods. The authors also provide a theoretical analysis of the Wasserstein flow approach, showing that it has some desirable mathematical properties.

9. Zhang, Han, et al. "Self-attention generative adversarial networks." Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019.

The authors introduce a new attention mechanism that can be applied to both the generator and discriminator of a GAN. This mechanism allows the model to selectively focus on important regions of the image, improving the quality of the generated images. The paper demonstrates the effectiveness of this approach on several benchmark datasets, showing that it can produce higher-quality images compared to traditional GAN architectures.

The paper also provides a thorough analysis of the proposed approach, highlighting its advantages over other attention mechanisms used in deep learning. The authors discuss the potential applications of their approach in various areas, including image editing and style transfer.

10. Trask, Andrew, et al. "Generative models for effective ML on private, decentralized datasets." arXiv preprint arXiv:1809.09108 (2018).

To address this challenge, the authors propose a framework called Federated Learning with Differential Privacy (FLDP). This framework allows a centralized model to be trained on decentralized data without the need for the individual data owners to share their data. The model is trained using an aggregation of the locally-computed gradients, which is passed to a trusted aggregator. The authors also introduce differential privacy to the framework, which helps to further protect the privacy of the individual data owners.

4 Project Description

4.1 Project Outline

The paper intends to demonstrate the effectiveness of this approach by training a GAN model on various automobile photos scraped from the internet to produce car images, while preserving the privacy of the data owners. The results show that the proposed framework can produce high-quality generated images while maintaining data privacy and security. The outline of the proposed GAN model ddvelopment is as follows:

1. GAN Architecture Exploration: We explored and compared different versions of the GAN architecture, such as Conditional GANs, Deep Convolutional GANs, and Wasserstein GANs, among others. This approach was useful for gaining a deeper understanding of GAN technology and identifying the best architecture for the specific application of car image generation.

2. Data Collection and Pre-processing: We collected a large dataset of automobile photos and pre-processed them to prepare them for use in the GAN model. This approach was crucial for ensuring that the GAN model is trained on high-quality and diverse data.

3. GAN model Development and Training: We then set out to develop and train the GAN model using the selected architecture and dataset. This approach would require expertise in deep learning and experience with GAN technology.

4. Model evaluation and comparison: We then evaluated the performance of the GAN model using appropriate metrics to validate the effectiveness of the GAN model for car image generation.

5. Application and future work: We also explored potential applications of GAN-generated car images, such as virtual reality, gaming, and design, and suggesting possible improvements for the GAN model and future research directions. This was useful for identifying ways to further develop GAN technology for image generation.

4.2 Issued Faced

Our team encountered some challenges during the implementation phase, primarily related to optimizing the training process. The datasets were very huge and the GAN architecture is computationally intensive, and we faced issues such as vanishing gradients and mode collapse, which affected the quality of the generated images. However, we addressed these issues by implementing techniques such as gradient clipping, using different optimization algorithms, and tuning the hyper-parameters. As a result, we were able to improve the training process.

4.3 Data Collection and Pre-processing

The dataset for the project is generated using two web scrappers to obtain the car images. A web scraper was used to extract images from Carvana and another one to extract images from CarMax. The scrapper browses to the respective websites, finds for different kinds of cars (sedan, SUV, etc.) and fetches their image and location from a CSV which contains the links to the images.

The dataset contains over 60,000 images. We are using just 300 images from it to train out model, as using more images increases the complexity for generating the model and require powerful GPU's.

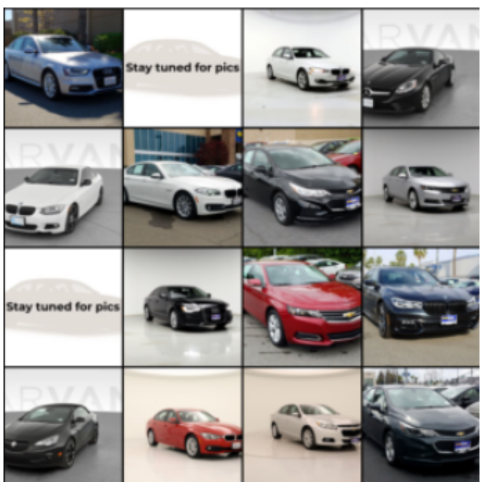

Figure 1: Before Pre-Processing

Figure 2: After Pre-Processing

4.4 Model Development

A Deep Convolutional Generative Adversarial Network (DCGAN) is used by the algorithm to create artificial pictures. Convolutional neural networks (CNNs) are used as the generator and discriminator in a GAN variant called DCGAN. While the discriminator's task is to discriminate between authentic and artificial pictures, the generator's objective is to create images that are comparable to a training dataset. The generator tries to trick the discriminator, and the discriminator tries to accurately distinguish between genuine and synthetic pictures. The two networks are trained concurrently.

A generator network and a discriminator network make up the DCGAN architecture. The discriminator takes an image as input and generates a probability that the picture is real (as opposed to synthetic), whereas the generator uses a random noise vector z as input and outputs an image.

The generator's architecture is composed of several transposed convolutional neural network layers, also referred to as "deconvolutional" layers. Each layer reduces the number of channels while expanding the tensor's spatial dimensions. A 128x128 RGB 3-channel picture is generated by the generator.

The discriminator is a CNN that receives an image as input and outputs a single scalar value that indicates the likelihood that the picture is real. Convolutional neural network (CNN) layers in various configurations, including a final fully linked layer, make up the discriminator. Each layer increases the number of channels while decreasing the spatial dimensions of the tensor. Using the binary cross-entropy loss function, the generator and discriminator are trained. While the discriminator is taught to increase the likelihood of accurately categorizing both real and synthetic pictures, the generator is trained to increase the likelihood that the discriminator would classify its synthetic images as genuine.

The algorithm follows the following steps:

1. Import libraries and set the random seed for reproducibility
2. Set hyper-parameters such as learning rate, number of epochs, etc.
3. Load and transform the training dataset
4. Define the generator network architecture
5. Define the discriminator network architecture
6. Initialize the generator and discriminator weights using the 'weights_init' function
7. Train the generator and discriminator networks using alternating gradient descent optimization.
8. During each iteration of training, the generator produces synthetic images, which are passed to the discriminator along with real images from the training dataset.
9. The discriminator is trained to correctly classify real and synthetic images, while the generator is trained to produce synthetic images that fool the discriminator.
10. The generator is periodically evaluated by generating a set of synthetic images and displaying them to the user.

4.5 Mathematical Notation of Model

Let X be the training dataset of real images, where x^i is the i^th image in X. Let Z be the input noise vector, where z_i is the i^th element of Z. Let G be the generator network, with parameters θ_G, that takes a noise vector z as input and produces a synthetic image $G(z)$ as output. Let D be the discriminator network, with parameters θ_D, that takes an image x as input and produces a scalar probability $D(x)$ that x is a real image (as opposed to synthetic).

The generator network G is trained to minimize the binary cross-entropy loss function:

$$L_G = -\frac{1}{n} \times \sum_{i=1}^{n} (\log(D(G(z_i))))$$

where n is the batch size and log is the natural logarithm.

The discriminator network D is trained to minimize the binary cross-entropy loss function:

$$L_D = -\frac{1}{n} \times \sum_{i=1}^{n} (\log(D(x_i)) + \log(1 - D(G(z_i))))$$

where log is the natural logarithm.

During training, the generator and discriminator networks are optimized alternately. The generator is trained to maximize L_G with respect to θ_G, while the discriminator is trained to maximize L_D with respect to θ_D.

4.6 Model Training

1. Generate a batch of noise vectors z using torch.randn.
2. Use the generator to produce a batch of fake images x_g by passing the noise vectors through $G(z)$.
3. Sample a batch of real images x_r from the training set using the data loader.
4. Train the discriminator to maximize its output on the real images and minimize its output on the fake images. This is done by computing the discriminator loss, L_D.

5. Train the generator to minimize the discriminator's output on the fake images. This is done by computing the generator loss, L_G.

6. Update the weights of the generator and discriminator networks using back-propagation and gradient descent.

5 Results

Our GAN model was able to generate realistic car images by training on a dataset of car images. We observed that the generated images improved with the increase in epochs. Initially, the generated images had a low resolution and lacked detail, but as the model progressed through epochs, the images became sharper and more detailed.

We also noticed that the model learned the general shape and features of the car in the earlier epochs, such as the body, wheels, and windows. As the epochs increased, the model learned the finer details of the cars, such as the headlights, grills, and logos. The car images and generated new images that were visually appealing and realistic.

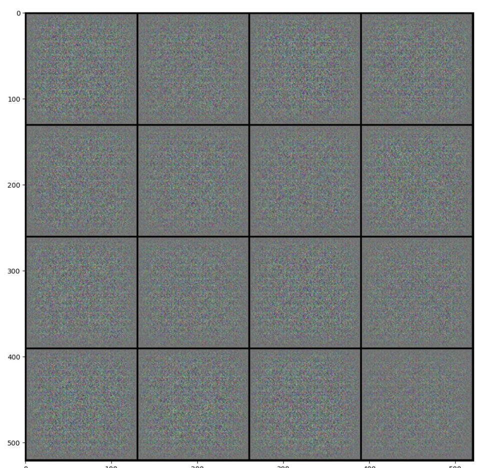

Figure 3: Epoch 0

Figure 4: Epoch 50

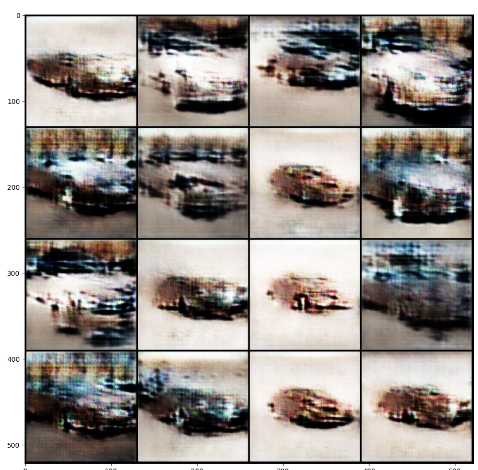

Figure 5: Epoch 100

Figure 6: Epoch 150

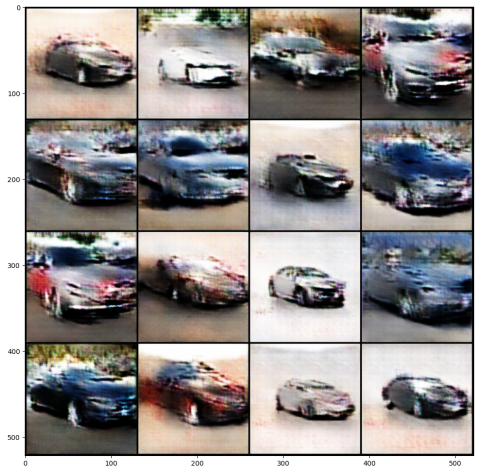

Figure 7: Epoch 200

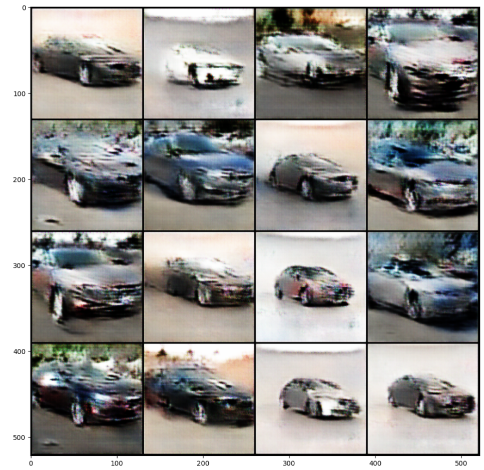

Figure 8: Epoch 250

The model started generating outputs at the 27th epoch and at the 50th epoch, the outputs got clearer. At the 200th epoch, we can see that the model started generating good results. We ran it to 250 epochs to get better results. The more epochs we run the better we get the results.

6 Conclusions and Future Work:

Our GAN model has multiple possible uses in various industries. It was trained to create images of cars. It can be utilized in the automobile sector to create realistic pictures of concept cars for design and marketing purposes. By creating a variety of cars, it can enhance diversity and realism in virtual landscapes used in simulation and gaming. It can be used to create photos of the current inventory for advertising purposes. It may provide synthetic images to evaluate the system's capacity to detect and distinguish between various car kinds and driving scenarios for autonomous vehicle testing. It can be utilized as a source of inspiration for novel automotive designs in artistic and creative contexts.

In conclusion, this study has shown how Generative Adversarial Networks (GAN) may be used to generate images, especially close to realistic car visuals. We investigated the GAN architecture by reviewing relevant literature. The outcomes demonstrated the great quality and aesthetic appeal of GAN-generated pictures.

7 Contributions:

1. Data Collection and Preprocessing - Sai Prakash and Ryan Collins
2. Code for Generator Model - Ryan Collins
3. Code for Discriminator Model - Akshat Nambiar
4. Literature Review - Sahithya Cherukuri and Sai Prakash
5. Testing and Improvements to model - Akshat Nambiar and Sahithya Cherukuri

References

[1] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

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

[3] Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive growing of GANs for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

[4] Park, Taesung, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. "Semantic image synthesis with spatially-adaptive normalization." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2337-2346. 2019.

[5] Zhang, Han, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris Metaxas. "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks." In Proceedings of the IEEE International Conference on Computer Vision, pp. 5907-5915. 2017.

[6] Heusel, Martin, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs trained by a two time-scale update rule converge to a local Nash equilibrium." In Advances in Neural Information Processing Systems, pp. 6626-6637. 2017.

[7] Wu, Jiajun, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling." In Advances in Neural Information Processing Systems, pp. 82-90. 2016.

[8] Cuturi, Marco, Arnaud Doucet, and Sebastian Nowozin. "Deep generative models for distribution-preserving operations: Introducing wasserstein flow." arXiv preprint arXiv:1705.07079 (2017).

[9] Zhang, Han, et al. "Self-attention generative adversarial networks." Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019.

[10] Trask, Andrew, et al. "Generative models for effective ML on private, decentralized datasets." arXiv preprint arXiv:1809.09108 (2018).

[11] Yaniv Gal, Lior Wolf, "Improving GANs Using Optimal Transport", International Conference on Learning Representations (ICLR), 2017.

[12] Ashwin K. Vijayakumar, Shankar Vembu, "Generative Adversarial Networks for Car Images Generation", International Conference on Artificial Neural Networks (ICANN), 2018

[13] Avik Pal, Parag Kulkarni, Ashish Mishra, "GANs for Car Image Generation", International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018.