applied sciences

MDPI

Article

LSUN-Stanford Car Dataset: Enhancing Large-Scale Car Image Datasets Using Deep Learning for Usage in GAN Training

Tin Kramberger [1,2,*] and Božidar Potočnik [2]

[1] Informatics and Computing Department, Zagreb University of Applied Sciences, 10000 Zagreb, Croatia
[2] Faculty of Electrical Engineering and Computer Science, University of Maribor, 2000 Maribor, Slovenia; bozidar.potocnik@um.si
* Correspondence: tin.kramberger@tvz.hr

Received: 25 June 2020; Accepted:15 July 2020; Published: 17 July 2020

Abstract: Currently there is no publicly available adequate dataset that could be used for training Generative Adversarial Networks (GANs) on car images. All available car datasets differ in noise, pose, and zoom levels. Thus, the objective of this work was to create an improved car image dataset that would be better suited for GAN training. To improve the performance of the GAN, we coupled the LSUN and Stanford car datasets. A new merged dataset was then pruned in order to adjust zoom levels and reduce the noise of images. This process resulted in fewer images that could be used for training, with increased quality though. This pruned dataset was evaluated by training the StyleGAN with original settings. Pruning the combined LSUN and Stanford datasets resulted in 2,067,710 images of cars with less noise and more adjusted zoom levels. The training of the StyleGAN on the LSUN-Stanford car dataset proved to be superior to the training with just the LSUN dataset by 3.7% using the Fréchet Inception Distance (FID) as a metric. Results pointed out that the proposed LSUN-Stanford car dataset is more consistent and better suited for training GAN neural networks than other currently available large car datasets.

Keywords: GAN dataset; car image dataset; Generative Adversarial Network; automotive image dataset; GAN neural network

1. Introduction

In recent years, the need for quality and large datasets has increased dramatically in the area of deep learning. Large high-quality datasets are of great importance to today's neural network training, because the data they contain reflect on the precision and accuracy of the output of the neural network. Training unsupervised neural networks such as Generative Adversarial Networks (GANs) has increased this requirement even further. Training GANs requires a specific dataset in terms of low intrinsic variation in poses, zoom levels, and backgrounds [1]. At this point, many datasets do not meet the training needs of GANs, given the amount and accuracy of data needed for successful training, and as such, produce low qualitative quality results on trained GANs, which can be seen in Section 6. Training GANs is an extremely dynamic process that requires diverse images and is very sensitive to every aspect of its training settings and training data. Current neural network models such as [1–3] need a large amount of data to avoid overfitting and train the GAN network properly. The GAN learns to distribute data from the dataset in such a way that the discriminator is trained to distinguish a sample from a model distribution. For example, the StyleGAN created by Karras et al. [1] has been trained on several different datasets: LSUN car, bedroom, and FFHQ datasets [1,4]. The automotive industry, with all the accompanying services, is one of the very broad application areas with a great

market share. A large and high-quality car dataset is, therefore, essential in this area. Observing the existing LSUN car dataset, one can see immediately that the data are not completely pruned, and that there is room for improvement as it contains various types of noise. An example is shown in Figure 1. This is simply a consequence of using Amazon's Mechanical Turk service [5] for data annotation. Amazon Mechanical Turk is a crowdfunding marketplace that outsources the process and jobs to a distributed workforce that performs these tasks virtually. As stated in [6], it has been observed that Mechanical Turks can be extremely imprecise when working with data, and that humans are not a good choice for classifying and annotating images.

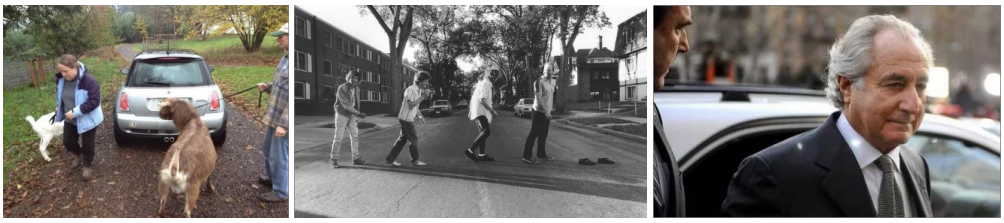

Figure 1. LSUN car dataset image noise examples [4].

This paper proposes a new LSUN-Stanford car dataset which is a union of the pruned and improved LSUN and Stanford car datasets. This new car dataset is aimed primarily for unsupervised neural network training, such as GAN training. Namely, our proposed dataset does not prescribe a division into training, validation, and testing sets, so in its original form and without modifications, it is not suitable for supervised training. It consists of an annotated and pruned LSUN car dataset [4] coupled with the annotated and pruned Stanford car dataset [7]. Our proposed dataset was constructed by filtering out the images where the car was the most salient object in the image. We achieved this by using Convolutional Neural Networks (CNNs) for object detection and cropping out the noisy parts of images, or discarding the image entirely if it was deemed unusable. Using that method, we have created a more refined car dataset intended for training GANs. The newly created dataset was then used to retrain the StyleGAN neural network with the same parameters as [1], and achieved superior Fréchet Inception Distance (FID) [8] compared to the original LSUN car dataset. The main motivation for this work was to create a better and more suitable large car dataset for training GAN neural networks. This dataset is publicly available, and is accompanied with all the required programming routines for its manipulation. A scientific contribution of this research is also the demonstration that the training of GAN networks is improved significantly by using our refined LSUN-Stanford car dataset.

This paper is organized as follows. Related work on existing car datasets is reviewed briefly in Section 2. Afterwards, the main highlights of the Generative Adversarial Network StyleGAN that is used in our experiments are presented in Section 3. There follows a detailed description of the creation and structure of the proposed LSUN-Stanford car dataset in Section 4. The experiments conducted using this new dataset are set out in Section 5. Section 6 reports the obtained qualitative and quantitative results. This paper is concluded in Section 7, where first, the benefits of the LSUN-Stanford car dataset are discussed and demonstrated, and finally, some future research directions are specified.

2. Similar Work and Existing Car Datasets

At this moment there are only a few specific datasets that exceed one million images, such as LSUN, Google Open Images, Tencent ML-Images, and ImageNet [7,9–11]. The majority of them are created for image classification and segmentation tasks. There are only a few datasets that meet the specific requirements for successful GAN training. Let us list the three main requirements, namely, low intrinsic variation in (I) poses, (II) zoom levels, and (III) backgrounds [1]. Karras et al. [1] have created the Flickr-Faces-HQ (FFHQ) dataset that contains 70,000 images of human faces, with a larger variation than the CelebA-HQ dataset [2] in terms of age, ethnicity, and background; and with better coverage of accessories, such as glasses, sunglasses, hats, and similar objects. All this was intended

specifically for the StyleGAN training, due to the possibility to add some specific style or feature (e.g., glasses) to the network. However, using the CelebA-HQ dataset for training pointed out that the preparation of a specific dataset in terms of low intrinsic variation in poses, zoom levels, and background yields better results in GAN training. CelebA-HQ offers much higher quality and covers a considerably wider variation than the existing high-resolution datasets [1].

The most popular datasets that include vehicles are KITTI [12], Stanford car dataset [7], Vehicle-1M [13], and the LSUN car dataset [4]. The KITTI object detection dataset contains 12,000 images of scenes with around 80,000 objects in total. The dataset was intended primarily for developing autonomous driving algorithms. Due to the high occlusion rate, it is not well suited for GAN training [12]. The Vehicle-1M dataset was created primarily for vehicle identification. It consists of 936,051 images taken from different traffic cameras in China, yet only 55,527 different vehicles are present, rendering it unsuitable for GAN training. Except for lack of vehicle diversity, the image quality is poor and of low resolution [13], and therefore inadequate for GAN training. Due to a large number of images, the LSUN car is one of the most popular datasets for GAN training. It suffers from high intrinsic variations in zoom levels, poses, and backgrounds of images. Additionally, the images can contain multiple cars; cars can also be occluded. Sample images from this dataset are depicted in Figure 1. The Stanford car dataset contains 16,185 images of 196 classes of cars. Each car class typically contains information about the make, model, and year [7]. It is much more pruned than the LSUN car dataset, but it is still affected by multiple car instances and noise in terms of watermarks on images, which can be seen in Figure 2.

Figure 2. Stanford car dataset image noise examples [7].

Karras et al. [1] have managed to generate images with better precision and quality using the original StyleGAN architecture and FFHQ dataset. However, the generated results were not representative when the original StyleGAN was trained with the LSUN car dataset [1]. This observation suggests that the pruned LSUN car dataset, coupled with pruned Stanford car dataset, could improve GAN neural network training.

3. Generative Adversarial Network StyleGAN

Generative Adversarial Networks were created in 2014 by Goodfellow et al. [3]. The GANs consist of two networks named Generator (G) and Discriminator (D). Both mentioned networks make GANs extremely complex and sensitive with respect to the (hyper)parameters. The reason is that G and D networks are based on a game theory and must be aligned perfectly [3,6]. The goal of GANs is to train a generator network $G(z;\theta^{(G)})$ that creates instances (in this case images) from data distribution, $p_{data}(x)$, transforming the noise vectors z into samples $x = G(z;\theta^{(G)})$. The letter z denotes the latent features of the images being generated, G is the generator, and $\theta^{(G)}$ is the neural network model. The training signal for G is provided by the Discriminator network $D(x)$. This network is trained to distinguish samples (images) from the distribution of the Generator $p_{data}(x)$ from the actual data. State-of-the-art GANs generate artificial or fake images of extremely high quality. It is practically impossible to distinguish such fake images from real images by observing just the visual image characteristics [1,2,6,14–17]. There are many variations of GANs that seek to improve training and model convergence on test data [6].

The StyleGAN neural network was used in an experimental part of this research. This state-of-the-art network is one of the latest GANs that achieves superior results with respect to the

FID metric [1]. The StyleGAN is the evolution of the progressive GAN [2]. It was implemented using the Tensorflow framework [18]. Similarly to the progressive GAN, the StyleGAN applies progression of the image size (resolution to some extent) during training. This means that the training starts by using smaller images, by which only layers in the generator that output this specific size of images are trained. At the same time, only layers with this specific image input size are trained in the discriminator. After 8.4 M images, the training continues by using the images of full 1024 × 1024 pixel (resolution) size from the dataset. This technique improves the performance of the training in terms of speed and stability of the GAN drastically [1,2]. The generator architecture of the StyleGAN is depicted in Figure 3. The traditional generator feeds the latent code only through the input layer, while the style generator maps the input to an intermediate latent space W, which controls the generator through adaptive instance normalization at each convolutional layer. Gaussian noise is added after each convolution. "A" stands for learned affine transform, and "B" applies learned per-channel scaling factors to the noise input. Most GANs use latent code that is provided to the generator through the input layer. However, the StyleGAN omits the input layer completely, and starts from a trained constant tensor instead (i.e., Const 4 × 4 × 512 in Figure 3) [1]. This network starts training by using a sample of 4 × 4 pixels and upsamples the image progressively to the maximum size of 1024 × 1024 pixels. The image sizes (resolutions) are denoted in the bottom right corner of each layer of the synthesis network (see Figure 3). The adaptive instance normalization (AdaIN) is used to apply the style transfer to the StyleGAN if style is needed [19]. The AdaIN is defined in Equation (1). It can be observed that each feature map x_i is normalized separately. The normalized feature map is then scaled and biased by using the appropriate style scalar components y_s and y_b. Denotation $\sigma(x)$ stands for normalized content input, while $\mu(x)$ denotes a shift. The AdaIN that receives the content x and a style y as inputs simply aligns the channelwise mean and variance of x to match those of y.

$$AdaIN(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \tag{1}$$

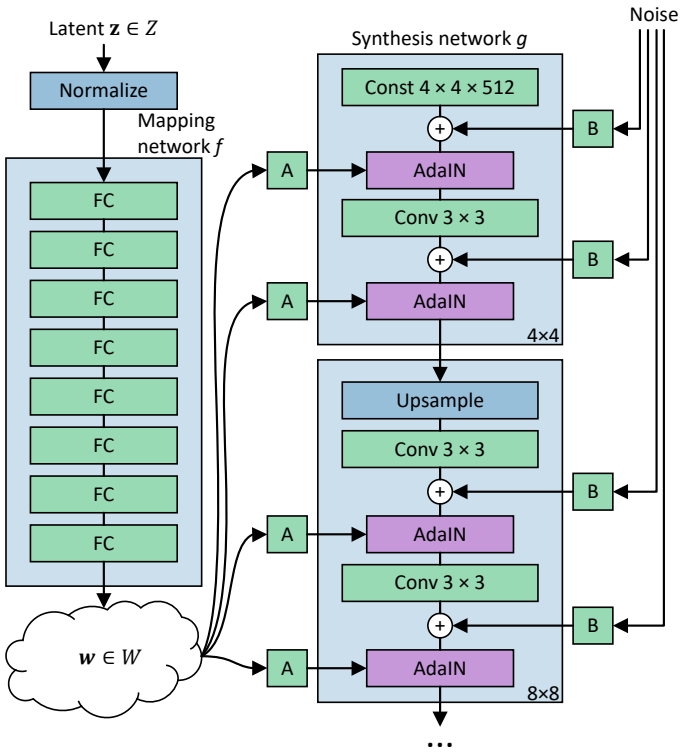

Figure 3. Generator architecture of the StyleGAN neural network [1].

Mapping network f consists of eight layers, while the synthesis network g consists of 18 layers [1]. The network has a total of 26.2M trainable parameters. The Discriminator network is the same as in [2], and consists mainly of replicated 3-layer blocks that are introduced one by one during the training. The structure of the Discriminator network is gathered in Table 1. Each layer starts with the convolution of a specific kernel size (denoted as Conv in Table 1), followed by the downsampling to the image size that corresponds to the upsampling of the generator network. All layers have leaky rectified linear unit activations (denoted as LReLU in Table 1) with $\alpha = 0.2$. The last layer is a fully-connected layer with the output size equal to 1. This layer returns a decision whether the image is real or fake.

Table 1. Discriminator architecture of the StyleGAN neural network [2].

Discriminator	Activation	Output Shape	Params
Input image	–	$3 \times 1024 \times 1024$	–
Conv 1×1	LReLU	$16 \times 1024 \times 1024$	64
Conv 3×3	LReLU	$16 \times 1024 \times 1024$	2.3k
Conv 3×3	LReLU	$32 \times 1024 \times 1024$	4.6k
Downsample	–	$32 \times 512 \times 512$	–
Conv 3×3	LReLU	$32 \times 512 \times 512$	9.2k
Conv 3×3	LReLU	$64 \times 512 \times 512$	18k
Downsample	–	$64 \times 256 \times 256$	–
Conv 3×3	LReLU	$64 \times 256 \times 256$	37k
Conv 3×3	LReLU	$128 \times 256 \times 256$	74k
Downsample	–	$128 \times 128 \times 128$	–
Conv 3×3	LReLU	$128 \times 128 \times 128$	148k
Conv 3×3	LReLU	$256 \times 128 \times 128$	295k
Downsample	–	$256 \times 64 \times 64$	–
Conv 3×3	LReLU	$256 \times 64 \times 64$	590k
Conv 3×3	LReLU	$512 \times 64 \times 64$	1.2M
Downsample	–	$512 \times 32 \times 32$	–
Conv 3×3	LReLU	$512 \times 32 \times 32$	2.4M
Conv 3×3	LReLU	$512 \times 32 \times 32$	2.4M
Downsample	–	$512 \times 16 \times 16$	–
Conv 3×3	LReLU	$512 \times 16 \times 16$	2.4M
Conv 3×3	LReLU	$512 \times 16 \times 16$	2.4M
Downsample	–	$512 \times 8 \times 8$	–
Conv 3×3	LReLU	$512 \times 8 \times 8$	2.4M
Conv 3×3	LReLU	$512 \times 8 \times 8$	2.4M
Downsample	–	$512 \times 4 \times 4$	–
Minibatch stddev	–	$513 \times 4 \times 4$	–
Conv 3×3	LReLU	$512 \times 4 \times 4$	2.4M
Conv 4×4	LReLU	$512 \times 1 \times 1$	4.2M
Fully-connected	linear	$1 \times 1 \times 1$	513
Total trainable parameters			23.1M

4. LSUN-Stanford Car Dataset

It was discovered in Section 2 that existing large car datasets do not meet the requirements for GAN training fully, especially in terms of image zoom level and pose. To the best of our knowledge, the most appropriate datasets for training GANs are currently the LSUN car dataset due to its large size, and the Stanford car dataset due to its unambiguousness. The LSUN car dataset consists of 5,520,753 car images but has many flaws in terms of noise and image accuracy (e.g., on some images there is not a single car, but rather some trucks, people, vans, etc.). Some of these problems are demonstrated in Figure 1. The Stanford car dataset consists of 16,185 car images which are much more accurate and have less noise, but the size of this dataset is inadequate for GAN training. Due

to the already large number of images available in the LSUN [4] and Stanford [7] car datasets, it was decided in this research to couple and prune both datasets in such a way that a merged dataset would be more suitable for the above-mentioned GAN needs. Serious shortcomings of both datasets are that images are often taken from online car adverts and unreliable pages, have poor backgrounds, and have the cars overlapping with other objects in the scene. All this restricts the GANs from being trained perfectly. A small amount of noise in the dataset is useful as it reduces the possibility of overfitting. However, a greater amount of noise simply reduces the training quality [20,21]. Some problematic images from the LSUN and the Stanford car datasets are depicted in Figures 1 and 2.

In the sequel of this research, we took images from the LSUN [4] and the Stanford car [7] datasets and joined them in one single dataset. The total number of images after merging both datasets was 5,536,938. Subsequently, pre-trained neural networks and deep learning methods were used to prune and annotate this new dataset. The most natural way to exclude unsuitable images from our new big dataset is to annotate images automatically using one of the existing state-of-the-art trained neural networks. In general, such networks have greater classification precision than humans [22,23]. Accordingly, the object detection techniques were utilized in order to prune and annotate this dataset. Based on our experience, we selected the MMDetection toolbox [24] for this task. This toolbox was created on the PyTorch framework, and represents a state-of-the-art architecture for detecting objects. For each detection within the image, this toolbox returns a mask and a bounding box around the object. The MMDetection toolbox supports many different backbones and methods [24]. The ResNet-101 [25] backbone and faster R-CNN [26] were chosen for our research due to their high performance and accuracy. The ResNet-101 model was pretrained on the Microsoft COCO dataset that contains 91 common object categories, including the car category [27]. After applying object detection to our new dataset, all objects in images were annotated by bounding boxes and classified in 91 categories (including a car category). All data about the bounding boxes were stored into the MySQL database. This database consists of object categories, (bounding) boxes, and images. Images can have multiple boxes of the same or different categories. The database scheme is shown in Figure 4.

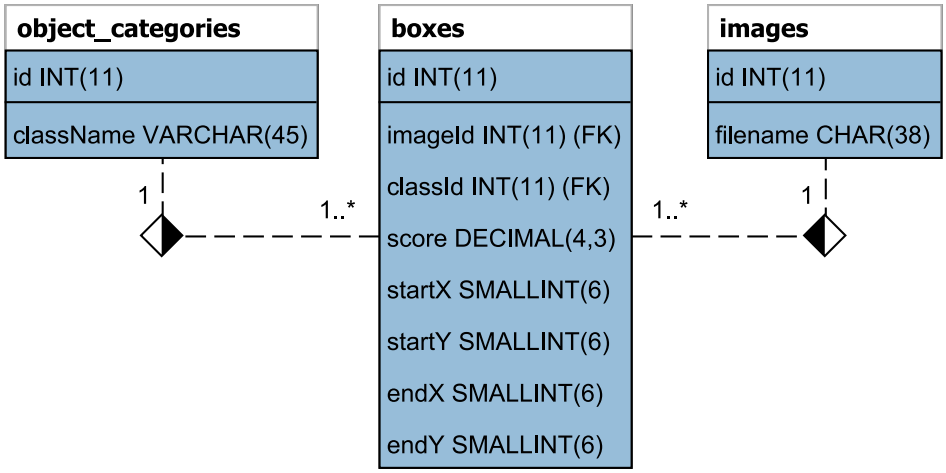

Figure 4. Database scheme for storing objects and bounding boxes within every image of the proposed LSUN-Stanford car dataset.

The table "images" contains all images from both the LSUN car and the Stanford car datasets, wherein filenames are retained from the original datasets. The table "object_categories" contains all 91 common object categories for which bounding boxes are created. The table "boxes" contains information about (bounding) boxes of a certain category on a selected image. Information about the (bounding) box, like start and end positions, and probability score, were extracted from the faster R-CNN object detection method.

Subsequently, the new joint dataset was pruned and some anomalies were removed. Let us describe this step more in detail. Only bounding boxes with cars were retained, and eventually,

bounding boxes with cars and drivers. We also requested that bounding boxes on the image do not overlap. All images were discarded that did not meet both criteria. This resulted in around two million pruned images. Afterwards, the images having multiple bounding boxes and multiple instances of cars in them were counted. It can be observed from Figure 5 that the new dataset obtained just by merging both existing datasets (i.e., in its initial form and without pruning), will not satisfy the training requirements for the GANs. Namely, many images contain more than one bounding box, and often more than one instance of a car in a single image. Besides, many images contain objects of other categories. The total number of images containing only one bounding box is 2,067,710 and the total number of images containing one bounding box of a car is 1,792,280. In order to increase the number of training images, we permit that images containing multiple cars that do not overlap are selected as well. Of course, such images should not contain objects other than cars. Exceptionally, an overlapping is allowed if a bounding box within the car is a bounding box of a person. We hypothesized that the person is a driver in such a situation. The result of pruning the initial LSUN and Stanford combined datasets in terms of number of bounding boxes can be seen in Figure 6. Sample images annotated with bounding boxes are depicted in Figure 7.

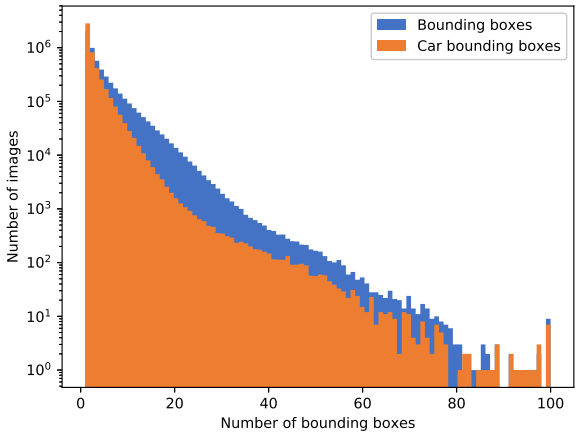

Figure 5. A number of images with respect to bounding boxes per image in the joined LSUN car dataset and the Stanford car dataset (without pruning). It can be seen clearly that in many images there are not just cars, but also many other objects (that represent noise for GAN training). The scale of the y-axis is logarithmic in order to present the distribution of data better.

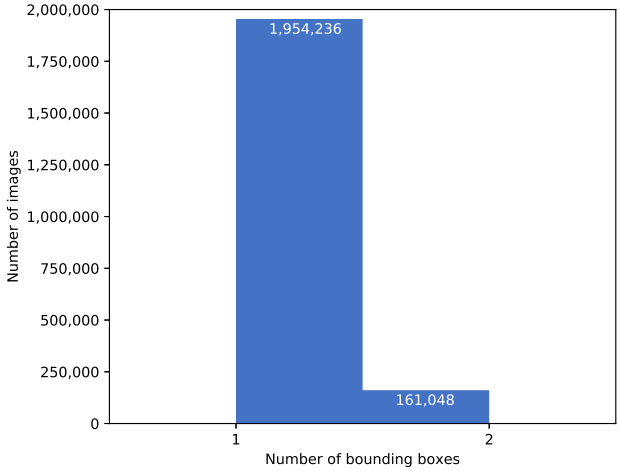

Figure 6. A number of images with respect to bounding boxes per image for our proposed LSUN-Standford car dataset. It can be seen clearly that, in the majority of images, there is just a car present, while in a few images there is a car with a driver (i.e., in images with two bounding boxes).

Figure 7. LSUN-Stanford car dataset: Images with appropriate bounding boxes.

Finally, the remaining images were cropped and resized in such a way that the aspect ratio was not altered (if possible) to achieve more accurate representations of images. Many authors, such as Karras et al. [1], have neglected this phase, and just resized the initial image, which could distort it. Consequently, the new LSUN-Stanford car dataset was constructed using the processing procedures described above. Our proposed car dataset with user instructions, MySQL database, and Python scripts for image manipulations is publicly available on the link https://github.com/Tin-Kramberger/LSUN-Stanford-dataset.

5. Computer Methods and Experimental Setup

We would like to demonstrate the benefits of the newly created LSUN-Stanford car dataset in this experimental part. Therefore, we retrained the StyleGAN neural network with our proposed dataset. Finally, we compared the obtained results with the state-of-the-art approach by Karras et al. [1]. The StyleGAN network should be trained by images of the size 512 $\times$ 384 pixels. For that reason, only images of that size or larger were considered from the LSUN-Standford dataset. The StyleGAN was trained in our experiment with the same hyperparameters as in [1]. That is why our results can be compared directly to results from [1]. The same progressive grow technique was implemented as that in [1]. This technique starts training the StyleGAN with images of size of 8 $\times$ 8 pixels, whereupon the size of training images grows progressively up to the size of 512 $\times$ 384 pixels.

6. Results

In our experiments, the StyleGAN network was implemented using the CUDA 10.0 library and Tensorflow 1.15.0, while PyTorch 1.2 was employed for the MMDetection toolkit. The object detection using MMDetection toolbox was performed on a computer with one Nvidia 1080 Ti graphics card. The estimated object detection speed using MMDetection toolbox is 10.9 frames per second [24]. However, with image preprocessing and postprocessing, the effective object detection speed was slightly lower in our experiments, i.e., around 10 frames per second. With approximately 5.5 million images in the LSUN and the Standford car datasets, a simple calculation points out that the object detection for both datasets took approximately six days and nine hours of processing time. Our implementation of the StyleGAN network was trained on a computer with two NVIDIA 1080 Ti GPUs, an AMD 1950X processor, and 32GB of RAM. The StyleGAN was trained by 46 million images. This training phase was completed in around 46 days and nine hours on our hardware.

Identically to Karras et al. [1], we utilized the Fréchet Inception Distance (FID) to assess the quality and efficiency of a trained GAN network. The Inception v3 neural network model [28] was utilized to calculate FID. Specifically, the last pooling layer prior to the output classification of images was used to capture computer vision-specific features of an output image. These activations were calculated for a collection of images. FID [8] is much more consistent than inception score [6] at estimating the

distance between a real and a generated image. Namely, FID applies real-world sample images for the comparison with synthetic images, unlike inception score, which uses only fake images to assess the quality of generated images.

The same evaluation protocol was employed as in Karras et al. [1]. We calculated the FIDs using 50,000 images drawn randomly from the LSUN-Standford dataset, and reported the lowest FID metrics during the training (all other images from this dataset were employed for the training). The number of images used for FID calculation is a parameter that can be set arbitrarily. However, it should be chosen appropriately to have FID values be as precise as possible, and such that the FID calculation would not take an extremely large amount of processing time. The calculation of the FID metric on 50,000 images was considered to be representative, because it was verified that the so-calculated FID metrics do not differ significantly from the FID metrics calculated on the entire dataset [6,29]. Figure 8 depicts how the FID metrics were changing during training of our StyleGAN network.

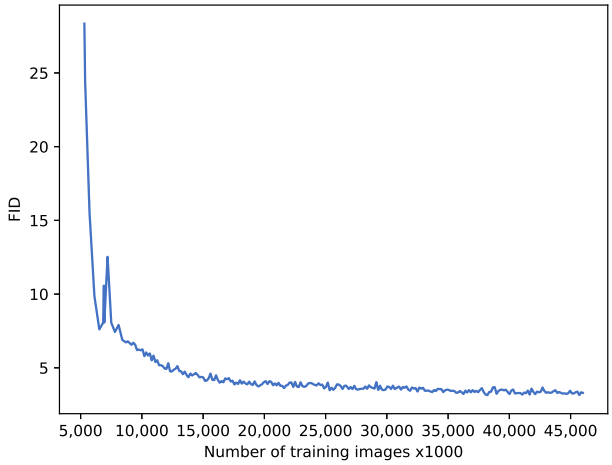

Figure 8. Change of FID metrics during training of the StyleGAN network. The proposed LSUN-Stanford car dataset was used. Axis x denotes the number of training images seen by the discriminator of the StyleGAN network. Some unstable behaviour can be noticed between 7M and 9M training images, which is a consequence of significant changes in the images' resolution and properties of the StyleGAN.

The final FID metric obtained after completing the training of our StyleGAN neural network on the pruned LSUN-Stanford Car Dataset was 3.15. For comparison, let us summarise the result if the SyleGAN neural network is trained just on the original LSUN car dataset. The calculated FID equalled 3.27 in this case [1]. Lower FID means better results. Around 3.7% improvement was observed when using the proposed LSUN-Standford car dataset for the StyleGAN training. It should be stressed that the same training protocol and StyleGAN hyperparameters were used in both experiments.

Let us also present some qualitative results. Figure 9 depicts generated images obtained by the generator network of our trained StyleGAN network. The visual quality is also exceptional for other generated images. For non-experts regarding cars, it is almost impossible to pinpoint the obvious flaws in generated cars. The process of our StyleGAN network training and generating cars for some sample images after each training epoch is demonstrated on the link https://youtu.be/NCuJAda7Qus. For comparison, we can inspect the generated cars in Figure 10. These images were generated by the StyleGAN, trained just with the original LSUN car dataset. The difference in visual quality between the two approaches or datasets is obvious.

Figure 9. Images of cars generated by our trained StyleGAN neural network using the proposed LSUN-Stanford car dataset.

Figure 10. Images of cars generated by the StyleGAN neural network trained on the original LSUN car dataset.

7. Discussion and Conclusions

The main intention of this paper was to introduce a public database of cars that is suitable for training GAN neural networks. We constructed a coupled dataset of cars using the LSUN car dataset and the Stanford car dataset. After coupling the datasets into one, it was pruned and the whole process of pruning was stored into a database. We provided the database structure and Python scripts which allow users to interact and export images from coupled datasets to their needs. The pruned and exported dataset was tested on the StyleGAN neural network. The results show a 3.7% lower FID compared to the StyleGAN trained just on the original LSUN dataset. These results can be explained by the fact that pruning made the dataset more consistent in terms of zoom levels, which yielded a better overall performance. One could argue that the better results were obtained because the Stanford car dataset was added to the LSUN dataset and then the StyleGAN was trained. However, it should be emphasized that we added less than 0.3% of images to the combined LSUN-Stanford car dataset. The original LSUN car dataset was already used to train other GAN architectures. Let us give some results for the comparison. It was noticed that the FID can vary largely with respect to the architecture. For example, FID was measured at 8.36 in [2] and at 2.66 in [30]. The metrics, especially the last one, were comparable to our results. In this experimental part, by maintaining the GAN architecture and just by modifying the database, we achieved a significant improvement of results. Therefore, we recommend to use this new combined LSUN-Stanford car dataset for GAN training.

The LSUN-Stanford car dataset leaves a lot of room for further improvements. The positions of observed objects in the image are extremely important when training a GAN neural network. Therefore, our first future work direction is to annotate the position of the car. Positioning can be done by annotating the headlights and tail lights, as well as the position of the wheels on each car that is already bounded by the box. In addition, it is also possible to annotate the car brand by using deep neural networks. This step could be very simple. For example, transfer learning could be utilized on the LSUN-Stanford car dataset, on which the weights of pre-trained neural network on the Stanford car dataset would be used. It should be stressed that the Stanford car dataset contains car types and brands.

Author Contributions: Conceptualization, T.K.; methodology, T.K.; software, T.K.; validation, B.P.; writing, T.K. and B.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Slovenian Research Agency (Contract P2-0041).

Acknowledgments: Tin Kramberger wishes to thank the University of Maribor for receiving a Doctoral Scholarship during this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4401–4410.
2. Karras, T.; Aila, T.; Laine, S.; Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
3. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Nice, France, 2014; pp. 2672–2680.
4. Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; Xiao, J. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *arXiv* **2015**, arXiv:1506.03365.
5. Buhrmester, M.; Kwang, T.; Gosling, S.D. Amazon's Mechanical Turk. *Perspect. Psychol. Sci.* **2011**, *6*, 3–5. [CrossRef] [PubMed]
6. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved techniques for training GANs. In Proceedings of the Advances in Neural Information Processing Systems 29, Barcelona, Spain, 5–10 December 2016; pp. 2234–2242.
7. Krause, J.; Stark, M.; Deng, J.; Li, F.-F. 3D object representations for fine-grained categorization. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 2–8 December 2013; pp. 554–561. [CrossRef]
8. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; Volume 2017, pp. 6627–6638.
9. Kuznetsova, A.; Rom, H.; Alldrin, N.; Uijlings, J.; Krasin, I.; Pont-Tuset, J.; Kamali, S.; Popov, S.; Malloci, M.; Kolesnikov, A.; et al. The Open Images Dataset V4. *Int. J. Comput. Vis.* **2020**. [CrossRef]
10. Wu, B.; Chen, W.; Fan, Y.; Zhang, Y.; Hou, J.; Liu, J.; Huang, J.; Zhang, T. Tencent ML-Images: A Large-Scale Multi-Label Image Database for Visual Representation Learning. *IEEE Access* **2019**, *7*, 172683–172693. [CrossRef]
11. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
12. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
13. Guo, H.; Zhao, C.; Liu, Z.; Wang, J.; Lu, H. Learning coarse-to-fine structured feature embedding for vehicle re-identification. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
14. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
15. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1947–1962. [CrossRef] [PubMed]
16. Denton, E.; Chintala, S.; Szlam, A.; Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 1, pp. 1486–1494.
17. Dzanic, T.; Witherden, F. Fourier Spectrum Discrepancies in Deep Network Generated Images. *arXiv* **2019**, arXiv:1911.06465.
18. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

19. Huang, X.; Belongie, S. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; Volume 2017, pp. 1510–1519. [CrossRef]
20. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017.
21. Bishop, C.M. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Comput.* **1995**, *7*, 108–116. [CrossRef]
22. Dodge, S.; Karam, L. A study and comparison of human and deep learning recognition performance under visual distortions. In Proceedings of the 2017 26th International Conference on Computer Communications and Networks, ICCCN 2017, Vancouver, BC, Canada, 31 July–3 August 2017. [CrossRef]
23. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; Volume 2015, pp. 1026–1034. [CrossRef]
24. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016, pp. 770–778. [CrossRef]
26. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
27. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Volume 8693 LNCS; Springer: New York, NY, USA, 2014; pp. 740–755. [CrossRef]
28. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016, pp. 2818–2826. [CrossRef]
29. Lucic, M.; Kurach, K.; Michalski, M.; Bousquet, O.; Gelly, S. Are Gans created equal? A large-scale study. In Proceedings of the Advances in Neural Information Processing Systems 31, Montréal, Canada, 3–8 December 2018; Volume 2018, pp. 700–709.
30. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. *arXiv* **2019**, arXiv:1912.04958.