Exploring Image Generation and Image Classification with Hybrid Datasets on the Task of Vehicle Identification

Matthias Gütebier

Institute of Cognitive Science
University of Osnabrück
49069 Osnabrück, Germany
mguetebier@uni-osnabrueck.de

Abstract. Hybrid datasets contain real data and synthetic data. In this paper, the application of synthetic data to an image classification task regarding vehicle identification is explored. After giving an overview on current research on image classification, the idea of hybrid datasets along with a terminology is introduced. The use of a deep convolutional generative adversarial network (DCGAN) for image generation is probed. Images generated from a DCGAN in combination with other images, real and synthetic, are then used as training data for a vehicle identification task.

Keywords: Image classification · Image generation · Vehicle identification · Hybrid dataset · Synthetic data

1 Introduction

This article deals with the topic of image classification, image generation, and using hybrid datasets for training neural networks. Image classification is a well established research field in computer vision. In vehicle identification, images of vehicles are categorized into classes of vehicles such as "car", "motorbike", or "bus". This use of image classification techniques offers a wide variety of applications in areas such as autonomous driving or traffic road monitoring. In more recent research, the idea of using artificially generated data in addition to real data during the training process of neural networks has come up. The aim of this paper is to give an overview on image classification techniques and on the idea of using synthetic data and hybrid datasets. Additionally, experiments on image generation and on image classification regarding vehicle identification are presented, where the practical use of synthetic data is explored.

Section *2 - Image Classification and Vehicle Identification* gives a short introduction into current research on image classification by presenting three articles which make use of different neural network architectures. In section *3 - Hybrid Datasets* the idea of artificially generated data is explored by introducing a terminology and explaining some benefits and pitfalls of using such data in training neural networks. Based on these introductions, section *4.1 - Image Generation* and *4.2 - Image Classification* present a series of experiments on image generation and image classification using hybrid datasets regarding vehicle identification. The article ends with a short conclusion in section *5 - Conclusion*.

For the experiments, a deep convolutional generative adversarial network was implemented and used for image generation. Experiments on image classification were done with an implementation of a convolutional neural network. The python scripts, trained models, and the datasets can be found in a GitHub repository[1].

2 Image Classification and Vehicle Identification

Image classification is a well established research field within Machine Learning and Computer Vision. Convolutional Neural Networks (CNNs), Transformers and other types of neural network architectures are used to classify images into given classes. In vehicle identification, image classification techniques are applied to images of different types of vehicles. These techniques are then supposed to identify a vehicle depicted in an image and classify the image into classes such as "car", "motorbike", "bus" and other classes. This allows for application in areas such as traffic monitoring, autonomous driving, and other image- or video-based systems regarding road traffic. With these areas of application, the challenges of vehicle identification become apparent. As road traffic takes place in all kinds of time and weather conditions, images of vehicles underlie a wide range of varying factors such as lighting, image quality, or external disturbances. Additionally, traffic cameras depict vehicles in different angles, and a variety of vehicle makes and models results in a variety of appearances in vehicles of the same class. To get an overview on current research in image classification and vehicle identification, three papers will be presented in the following paragraphs.

In their paper on image classification, Yuan et al. [10] apply the Transformer architecture known from Natural Language Processing to Computer Vision, consisting of an encoder model "T2T" and a decoder model "T2T-ViT backbone" [10, p. 562]. The encoder "T2T" splits an image into a sequence of tokens to model local information structures. Based on such a sequence of tokens, the decoder "T2T-ViT backbone" then predicts a label for the image. Yuan et al. [10] report experiments on an ImageNet [3] dataset with 1.3 million training images and 50.000 validation images. During training, the data augmentation techniques *Mixup* and *Cutmix* were used to alter training images. *Mixup* generates a new image by combining random image pairs from the training data, and *Cutmix* combines images by patching regions from one image into another image. Regarding their experiments on general image classification, Yuan et al. [10] report a classification accuracy of up to 83%.

An approach to car identification is presented by Corrales et al. [2]. They used a pretrained Residual Neural Network (RNN) from ImageNet [3] and finetuned it for classifying cars into different car models. RNNs are a type of CNN using

[1] Link to the GitHub repository: `https://github.com/guetmatt/Hybrid-Data-AI`

skip connections to avoid the degradation problem. Further details can be found in the original paper on RNNs by He et al. [4]. Corrales et al. [2] carried out experiments using a subset of the CompCars dataset [9]. They also use several data augmentation techniques on the training data such as horizontally flipping an image, introducing noise (*Salt & Pepper, Speckle, Poisson*), blurring an image or augmenting the colors by color casting or color jittering. In their experiments on car model identification, Corrales et al. [2] report an accuracy of up to 97%.

In another approach to image classification focusing on mobile and resource constrained environments, Sandler et al. [6] present MobileNet. This is a CNN with two layers, specifically designed to run on resource constrained devices. While performing experiments on a general ImageNet classification task without any kind of data augmentation during training, Sandler et al. [6] report an accuracy of up to 74%.

3 Hybrid Datasets

The previous section *2 - Image Classification and Vehicle Identification* provided an overview on current research on image classification and vehicle recognition. While two of the studies presented used data augmentation techniques during training ([2,10]), no usage of artificially generated data was described. To get a better understanding of the distinctions between augmented data, synthetic data, and other terms, the now following section gives a general overview on the topic of hybrid datasets.

To get a more precise idea of the topic, a well defined terminology is required. Regarding data samples, Wachter et al. [8] distinguish between *real data*, *synthetic data*, *augmented data*, and *partially synthetic data*. While *real data* is defined as "information collected from the physical world", *synthetic data* "is all artificially generated and transformed data" [8, p. 439]. *Augmented data* is understood as a subclass of *synthetic data*, being defined as "transformations of real data samples, which can be basic modifications, like cropping and rotation, as well as more complex transformations, such as those achieved through generative AI" [8, p. 439]. Therefore, images generated by a generative artificial intelligence are categorized as *augmented data* within the class of *synthetic data*, while images taken from simulations or created by hand drawing are seen as *synthetic data* without being *augmented data*. With *partially synthetic data*, the last subclass is an intersection defined as data samples that "possess real and synthetic attributes; one part originates from real sources, while the other is synthetically generated" [8, p. 439]. Widening their lens, Wachter et al. [8] then turn from data samples to datasets. Regarding datasets, they distinguish between *hybrid datasets* and *augmented datasets*. Based on the definitions made regarding data samples, *hybrid datasets* are defined as containing "both real and synthetic samples; the latter are not augmented data samples" [8, p. 440]. A dataset that contains real and augmented samples therefore is an *augmented dataset*, but if a dataset also contains synthetic samples (in addition to real and augmented samples) it is labeled as a *hybrid dataset*.

With these definitions in mind, some context on the use of synthetic data is needed. Two main concepts illustrate the relevance and possible difficulties of using hybrid datasets as training data, namely the *data problem* and the *reality gap*. With regard to the data needed to train an artificial intelligence, the *data problem* is defined by Wachter et al. as the "significant costs" that come with "Acquiring, creating, preparing and labeling this data" [8, p. 437]. Other aspects of the *data problem* are data sparsity and class imbalance in areas such as the agricultural sector [8]. Synthetic data and hybrid datasets can be a solution to the *data problem*, but have to be used carefully due to the resulting *reality gap*. Using only synthetic data as training data "causes a significant drop in prediction performance when models [...] are deployed to real-world scenarios" [8, p. 440], due to a lack of realism in synthetic data. Hybrid datasets, combining real data and synthetic data, can be a solution to the *data problem* as well as the *reality gap*, bridging the gap to the real world through real data, while enhancing the diversity of the dataset with synthetic data. Wachter et al. report a "data ratio of 10% real data and 90% synthetic data" [8, p. 440] in scenarios where very limited amounts of real data are available. Additionally, even when a large amount of real data is available, "introducing hybrid data continues to enhance model prediction performance" [8, p. 440].

4 Experiments

As the aim of this paper is to study the use of hybrid datasets as training data in the field of image classification, two kinds of experiments have been conducted. Subsection *4.1 - Image Generation* provides a short introduction on the use of deep convolutional generative adversarial networks for image generation and presents an experiment to generate synthetic data for later experiments on image classification and vehicle identification. These experiments are presented in subsection *4.2 - Image Classification.*

4.1 Image Generation

Radford et al. [5] introduced deep convolutional generative adversarial networks (DCGANs), a neural network architecture that can be used for generating images. Following the terminology introduced by Wachter et al. [8], the generated images from DCGANs fall into the subclass of augmented data within the class of synthetic data, as they are a product of generative artificial intelligence. Therefore, the term *augmented images* will be used from here on to refer to images generated by DCGANs.

DCGANs are a model architecture combining CNNs and generative adversarial networks (GANs). GANs can generally be used for generating images. They consist of two neural networks, namely the generator and the discriminator. Based on training data and random noise, the generator generates images, while the discriminator has to differentiate between real and generated images. During the training process, these two networks compete with each other, improving the

GAN. In their paper, Radford et al. introduce several architectural constraints to stabilize GANs, while the unsupervised training process stays the same, resulting in the DCGAN architecture [5, p. 3]. Radford et al. provide evidence "that adversarial networks learn good representations of images for supervised learning and generative modeling" [5, p. 9], showing one of the improvements of DCGANs.

To generate augmented images for the experiments presented in this paper, a basic DCGAN was implemented by following and customizing a tutorial[2] given by TensorFlow[3] [1]. The model was trained on the *Car vs Bike Classification Dataset*[4], consisting of 2000 real images of motorbikes and 2000 real images of cars and licensed under CC0: Public Domain[5]. For the purpose of this paper, the images have been resized to 112x112 pixels and are loaded in color mode grayscale. Firstly, the DCGAN was trained for 200 epochs with batches of 200 training images from the car images of the mentioned dataset. After training, 240 images of cars were generated. Figure 1 shows examples of generated images during different training epochs. Figure 2 shows a sample of the 240 generated images of cars. Due to the low quality of generated images and due to time and resource constraints, it was decided to not train the DCGAN further or generate more images of cars or motorbikes. A proof of concept regarding the possibility of generating augmented data with a DCGAN has been shown and later experiments on the use of hybrid datasets will include additional generated images from already available datasets. The bad quality of the generated images likely stems from a combination of training in an environment with limited computational resources and using a training dataset that consists of a fairly low amount of images.

[2] Tutorial: `https://www.tensorflow.org/tutorials/generative/dcgan`, last visited on September 1st, 2024

[3] Tensorflow: `https://www.tensorflow.org/`

[4] `https://www.kaggle.com/datasets/utkarshsaxenadn/car-vs-bike-classification-dataset`, last visited on September 1st, 2024

[5] License CC0: Public Domain: `https://creativecommons.org/publicdomain/zero/1.0/`

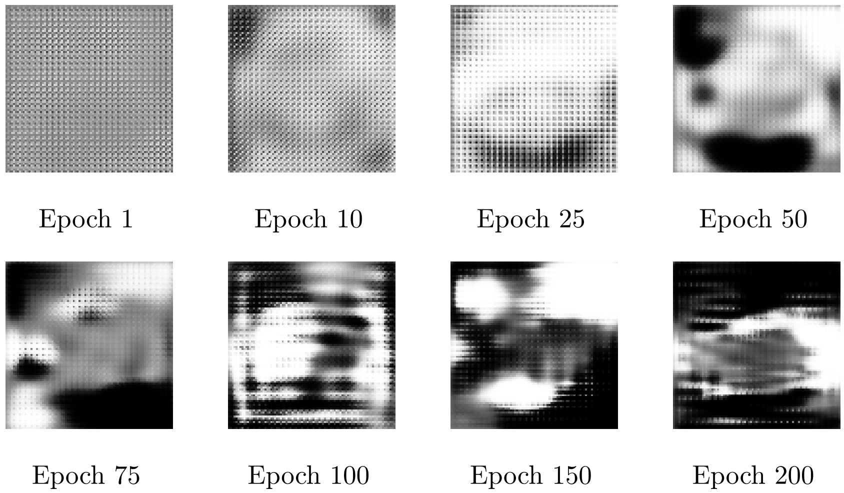

Fig. 1: Examples of generated car images after different training epochs

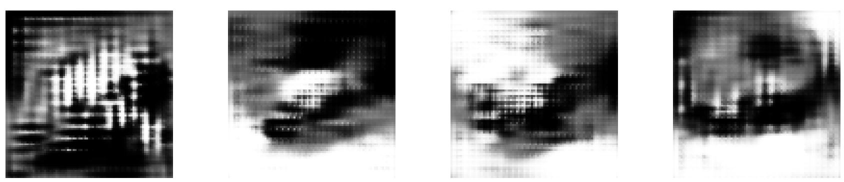

Fig. 2: A sample of generated car images after 200 training epochs

4.2 Image Classification

This section now presents a series of experiments on vehicle identification using image classification. To classify images, a standard convolutional neural network (CNN) was implemented, based on a tutorial[6] from TensorFlow [1]. The model from this tutorial has been adjusted to fit the datasets used for the vehicle identification task and some additional parameters have been changed. It consists of three convolutional layers and a fully connected layer at the end. All images are loaded in color mode grayscale and have been resized to 112x112 pixels. To avoid overfitting during training, a method of *early stopping*[7] was used with varying settings of the parameter *patience*. Early stopping causes the training process to stop when a certain training metric does not improve for a number of epochs. The number of epochs is determined by the patience parameter, and the training metric monitored in this case is the *validation loss*. Another technique applied to prevent the model from overfitting was *dropout*, randomly setting inputs during training to 0, adding more variety to the data.

The series of experiments can be split into two kinds of experiments with four different training approaches for each of these kinds. All experiments are aimed at exploring the impact of synthetic and augmented training data on model performance. Experiments 1.1 to 1.4 are concerned with distinguishing cars from motorbikes, or more generally distinguishing four-wheel vehicles from two-wheel vehicles. As people who use two-wheel vehicles are more exposed to the dangers of road traffic than people in four-wheel vehicles, such a distinction is important for the safety of two-wheel vehicle drivers. In experiments 2.1 to 2.4, the idea of a more fine-grained vehicle identification is tested by adding busses to the set of classes. For training, validation, and testing, a combination of several datasets was used. As this paper focuses on hybrid datasets, several kinds of synthetic data have been tested. The synthetic data used during these experiments consists of car images generated in the process described in section *4.1 - Image Generation*, the already available *Synthetic Image Dataset*[8] (License: CC BY-NC-SA 4.0[9]) and the application of data augmentation techniques. These data augmentation techniques were *RandomFlip*[10], *RandomRotation*[11], and *RandomZoom*[12], randomly flipping, rotating or zooming in on an image. Overall, the datasets used

[6] Tutorial: `https://www.tensorflow.org/tutorials/images/classification`

[7] Tensorflow early stopping: `https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping`

[8] Synthetic Image Dataset: `https://www.kaggle.com/datasets/zarkonium/synthetic-image-dataset-cats-dogs-bikes-cars/data`

[9] License CC BY-NC-SA 4.0: `https://creativecommons.org/licenses/by-nc-sa/4.0/`

[10] RandomFlip: `https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomFlip`

[11] RandomRotation: `https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomRotation`

[12] RandomZoom: `https://www.tensorflow.org/api_docs/python/tf/keras/layers/RandomZoom`

for training and testing are combined from the datasets *car-bike*[13], *vehicles*[14], *synthetic*[15] [7], and *augmented*. A detailed overview on the datasets is given by table 1.

Dataset	**No. of images**	**Type of images**	**License**
car-bike	2000 motorbikes 2000 cars	real data	CC0: Public Domain
vehicles	1354 motorbikes 1023 cars 1672 busses	real data	CC0: Public Domain
synthetic	2500 bikes 2500 cars	synthetic, generated in Unity	CC BY-NC-SA 4.0
augmented	240 cars	augmented, generated by DCGAN	self-generated

Table 1: Datasets

[13] dataset *car-bike* : `https://www.kaggle.com/datasets/utkarshsaxenadn/car-vs-bike-classification-dataset`

[14] dataset *vehicles*: `https://www.kaggle.com/datasets/mrtontrnok/5-vehichles-for-multicategory-classification`

[15] dataset *synthetic*: `https://www.kaggle.com/datasets/zarkonium/synthetic-image-dataset-cats-dogs-bikes-cars/data`

Experiment 1 – Cars vs. Bikes For experiments 1, 90% of the images from the *vehicles* dataset and the *car-bike* dataset were used for each class in training and validation. Additionally, 240 car images were used from the *augmented* dataset, and 260 car images as well as 300 bike images were used from the *synthetic* dataset. For testing, the remaining 10% of the images from the *vehicles* and the *car-bike* dataset were used. Table 2 provides a more detailed overview on the training data. A sample of the training data can be seen in figure 3. The performance of the trained models was tested on real data from the *vehicles* and *car-bike* datasets, resulting in 323 car images and 335 bike images. Table 3 shows detailed testing results as well as the setting of the *patience* parameter for early stopping and the epoch after which the training process stopped due to early stopping.

Fig. 3: A sample of the training data for experiments 1.1 to 1.4

Exp.	**Real data** datasets: *car-bike* & *vehicles*	**Synthetic data** datasets: *synthetic* & *augmented*	**Data augmentation applied?**	**Total training data**
Exp. 1.1	2699 cars 3017 bikes	None	No	2699 cars 3017 bikes
Exp. 1.2	2699 cars 3017 bikes	None	Yes	2699 cars 3017 bikes
Exp. 1.3	2699 cars 3017 bikes	240 cars *(augmented)*	Yes	2939 cars 3017 bikes
Exp. 1.4	2699 cars 3017 bikes	240 cars *(augmented)* 260 cars *(synthetic)* 300 bikes *(synthetic)*	Yes	3199 cars 3317 bikes

Table 2: Training and validation data for experiments 1.1 to 1.4

Exp.	**Early stopping patience**	**Early stopping epoch**	**Accuracy overall**	**Recall classwise**	**Precision classwise**
Exp. 1.1	3	7	0.9027	cars: 0.9195 bikes: 0.8865	cars: 0.8865 bikes: 0.9195
Exp. 1.2	3	5	0.8799	cars: 0.8607 bikes: 0.8985	cars: 0.8910 bikes: 0.8699
Exp. 1.3	3	15	0.8891	cars: 0.8452 bikes: 0.9313	cars: 0.9223 bikes: 0.8619
Exp. 1.4	5	16	0.8815	cars: 0.8653 bikes: 0.9254	cars: 0.9153 bikes: 0.8539

Table 3: Test results for experiments 1.1 to 1.4

Experiment 2 – Cars vs. Bikes vs. Busses For experiments 2.1 to 2.4, only images from the datasets *vehicles*, *augmented*, and *synthetic* were used, as the car and bike images from the *car-bike* dataset would have caused a strong difference in class size compared to the new class *bus*. Again, 90% of the data for each class was used for training and validation, while the remaining 10% were used for testing. As the *vehicles* dataset offers more images of busses than images of cars or bikes, the synthetic images of cars and bikes were used as a measure against data imbalance. See table 4 for further details. Figure 4 shows a sample of the training data. The test dataset for all trained models in these experiments was a subset of the *vehicles* dataset. It contained 167 bus images, 101 car images, and 135 bike images. For detailed test results and parameters for the early stopping method during training, see table 5.

Fig. 4: A sample of the training data for experiments 2.1 to 2.4

Exp.	**Real data** dataset: *vehicles*	**Synthetic data** datasets: *synthetic & augmented*	**Data augmentation applied?**	**Total training data**
Exp. 2.1	1505 busses 922 cars 1219 bikes	None	No	1505 busses 922 cars 1219 bikes
Exp. 2.2	1505 busses 922 cars 1219 bikes	None	Yes	1505 busses 922 cars 1219 bikes
Exp. 2.3	1505 busses 922 cars 1219 bikes	240 cars *(augmented)*	Yes	1505 busses 1162 cars 1219 bikes
Exp. 2.4	1505 busses 922 cars 1219 bikes	240 cars *(augmented)* 260 cars *(synthetic)* 300 bikes *(synthetic)*	Yes	1505 busses 1422 cars 1519 bikes

Table 4: Training and validation data for experiments 2.1 to 2.4

Exp.	**Early stopping patience**	**Early stopping epoch**	**Accuracy overall**	**Recall classwise**	**Precision classwise**
Exp. 2.1	7	12	0.8015	busses: 0.7545 cars: 0.6831 bikes: 0.9481	busses: 0.9474 cars: 0.69 bikes: 0.7529
Exp. 2.2	7	28	0.8387	busses: 0.8802 cars: 0.6931 bikes: 0.8963	busses: 0.84 cars: 0.7692 bikes: 0.8832
Exp. 2.3	7	25	0.8238	busses: 0.8204 cars: 0.7822 bikes: 0.8593	busses: 0.9073 cars: 0.6639 bikes: 0.8722
Exp. 2.4	10	30	0.7841	busses: 0.8563 cars: 0.4752 bikes: 0.9259	busses: 0.8125 cars: 0.7059 bikes: 0.7862

Table 5: Test results for experiments 2.1 to 2.4

5 Conclusion

Overall, the results achieved in the experiments on vehicle identification are decent, considering constraints of the computing environment and the low amount of data used for training and testing. As image classification is a well established research field, the methods and architectures available for this task deliver a good baseline performance. For higher accuracy scores, it is likely that more training data would have been needed. Additionally, the training dataset could have been more fleshed out considering factors such as the quality of images, variety of angles, and reduction of noise.

In the first set of experiments 1.1 to 1.4, the best accuracy was achieved in the first experiment, using only real data without any data augmentation or synthetic data. Interestingly, applying data augmentation techniques to the real data led to a slightly lower accuracy score in experiment 1.2. Important to note here are the early stopping epochs being quite low, with experiment 1.1 stopping training after epoch 7 and experiment 1.2 stopping even earlier after epoch 5. This is a sign of the model overfitting, likely due to low amount of training data. Applying data augmentation techniques to the real data and adding augmented and synthetic images in experiments 1.3 and 1.4 led to a slight decrease in accuracy compared to experiment 1.1, but the amount of training epochs increased to 15 and 16. Although the high accuracy scores still suggest overfitting due to low amount of training and test data, adding augmented and synthetic data seems to have a positive effect on the models ability to generalize. In this case, creating a hybrid dataset seems to be helpful in order to diversify and expand the training dataset. This effect is also achieved when adding only the augmented images from the self-generated *augmented* dataset to the training data, although the recall regarding car classification is the lowest in experiment 1.3 out of all experiments 1.1 to 1.4. As the images from the *augmented* dataset are labeled as car images while not being more than noisy images, this can be seen as adding noise to the training dataset, improving precision on cars at the cost of slight precision decrease on bikes.

Regarding the second set of experiments 2.1 to 2.4., the highest accuracy score was achieved in experiment 2.2 with real data and data augmentation techniques, although all experiments 2 achieve a reasonable accuracy. As early stopping epochs are higher than in experiments 1, the risk of the model overfitting seems to be lower. Note that training epoch 30 (stopping epoch in experiment 2.4) was the maximum amount of possible epochs in the training setup. Precision regarding busses is quite high in experiment 2.1, with the data augmentation techniques applied in experiment 2.2 nicely leveling out precision and recall regarding busses and bikes. This effect might be due to the training dataset consisting of roughly 20% more bus images compared to bike images. As the training contains even less car images, precision and recall on car classification is lowest out of all three classes. Surprisingly, the addition of augmented and synthetic data to the training dataset did not improve overall performance. Especially car classification plummeted to a recall score of 0.4752, although precision slightly increased in experiment 2.4 compared to 2.3. This could be a sign of overfitting in experiment

2.1 and 2.2, as the test data did not change and is quite similar to the training data in 2.1 and 2.2. By adding augmented and synthetic data to the training data, the model gains generalizability at the cost of performance on the test data. In this case, synthetic data was not able to smooth out the class imbalance in the training data set. Overall, the results suggest that hybrid training datasets led to better generalizability in models, although test performance was slightly worse than training with real data only. More exploration of hybrid datasets and the ratio of real data to synthetic or augmented data has to be done, as hybrid datasets seem to be promising for research fields dealing with low amounts of real data.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. & Zheng, X.: TensorFlow: a system for large-scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265–283 (2016)
2. Corrales, H., Llorca, D. F., Parra, I., Vigre, S., Quintanar, A., Lorenzo, J., & Hernández, N.: CNNs for fine-grained car model classification. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) Computer Aided Systems Theory–EUROCAST 2019: 17th International Conference, Las Palmas de Gran Canaria, Spain, February 17–22, 2019, Revised Selected Papers, Part II, pp. 104–112. Springer International Publishing (2020). https://doi.org/https://doi.org/10.1007/978-3-030-45096-0_13
3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp. 248–255, Ieee (2009)
4. He, K., Zhang, X., Ren, S., & Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778 (2016)
5. Radford, A., Metz, L., & Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434. (2015)
6. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520 (2018)
7. Valtchev, S.Z., Wu, J.: Domain randomization for neural network classification. J Big Data**8**, 94 (2021). https://doi.org/10.1186/s40537-021-00455-5
8. Wachter, P., Kruse, N., Schöning, J.: Synthetic fields, real gains. In: 44. GIL-Jahrestagung, Biodiversität fördern durch digitale Landwirtschaft, pp. 437–442, Gesellschaft für Informatik, Bonn (2024)
9. Yang, L., Luo, P., Change Loy, C., & Tang, X.: A large-scale car dataset for fine-grained categorization and verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3973–3981 (2015)
10. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z. H., Tay, F. E., Feng, J. & Yan, S.: Tokens-to-Toekn ViT: Training Vision Transformers From Scratch on ImageNet. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 558–567 (2021)