The Synthetic Data Application in the UAV Recognition Systems Development

Diana Duplevska
Institute of Electronics and Computer Science
Riga, Latvia
diana.duplevska@edi.lv

Vladislavs Medvedevs
Institute of Electronics and Computer Science
Riga, Latvia
vladislavs.medvedevs@edi.lv

Daniils Surmacs
Institute of Microwave Engineering and Electronics
Riga Technical University
Riga, Latvia
daniils.surmacs@rtu.lv

Arturs Aboltins
Institute of Microwave Engineering and Electronics
Riga Technical University
Riga, Latvia
aboltins@rtu.lv

***Abstract*—The increasing popularity and accessibility of unmanned aerial vehicles (UAVs) presents both opportunities and challenges. On the one hand, UAVs has a wide range of civilian, industrial, and military applications. On the other hand, the popularity of UAVs can lead to illegal or dangerous usage. Thus, the development of UAV recognition systems is crucial for ensuring safety and security. However, collecting and labeling large amounts of real-world data for training these systems can be time-consuming and labor-intensive.**

In this study, we propose a methodology, which can help to accelerate the development of new UAV recognition systems. This work demonstrates the effectiveness of training a neural network using a combination of real-world and synthetic data that can achieve similar performance to a network trained on real-world data only.

***Index Terms*—Neural networks, Convolutional neural networks, Artificial neural networks, Synthetic data, Generative adversarial networks.**

I. Introduction

The count of illegal and dangerous UAV use cases is growing together with this technology's popularity and availability. And technologies of prevention and countermeasures should be developed at the same pace.

The most basic of these systems utilize a single detection approach, such as visual-based, sound-based, radar-based, thermal-based, or radio frequency radio frequency (RF)-based detection. [1]–[5] These approaches are intended to identify UAVs based on their physical characteristics – image, sound, radar reflections, thermal signatures, or radio frequency signals, respectively.

More advanced systems additionally detect different drone patterns in order to recognize the UAVs. This approach allows for assessing the level of threat. But the main challenge for this approach is the correct pattern definition, which slows down the development of these systems.

The modern solution for this problem is artificial neural network (ANN)-based system that automatically defines drone patterns by training on the UAV representing dataset. The significant problem in the artificial intelligence (AI) and macine learning (ML) system design is training dataset collection and its management. Usually, it takes about 80% of the time spent on system development [6].

Machine learning algorithms require large amounts of structured data for training and testing. For example, to solve machine vision problems, ImageNet [7] datasets are used – in this database are more than 14 million images, divided into thousands of categories. Using such a high-quality and sorted dataset allows to achieve a more accurate AI model. Algorithms using ImageNet make mistakes in identifying an object in photographs in only 3.75% of cases. In comparison, humans make mistakes in 5% of cases [8].

But it is impossible to form datasets like ImageNet for each task. If only because the data in them is marked, checked, and categorized manually. Also, in some cases, real data may be closed and inaccessible to developers due to data protection, which leads to the privilege of selling and distributing data from owners and duplicating available information in public access [9].

Synthetic data can help to solve these difficulties. They are computer generated but look similar to real ones.

Since this paper is devoted to visual-based detection and recognition approach, the dataset is represented as an image collection. So, another issue of the synthetic dataset could be the lack of realism of the generated data [10].

The application of photorealistic synthetic data in the training of UAV detection systems is crucial for the development of accurate and precise UAV recognition systems. The realism of the synthetic data allows the neural network to learn realistic patterns and features of UAVs, improving the accuracy of the model in real-world scenarios. The diversity and realism of synthetic data also allow the convolutional neural network (CNN) to adapt to various lighting conditions, weather conditions, and camera angles, making the UAV detection system more robust.

The rest of this paper is structured as follows:

- Section II describes related work on similar technology;
- Section III describes the experimental setup of synthetic data influence on the UAV recognition system;
- Section IV describes an experimental results;
- Section V describes a discussion on this work.

979-8-3503-1178-5/23/$31.00

II. Related Works

Recent studies have investigated the use of synthetic data in the design of UAV recognition systems, with a growing number of works addressing the problem due to its relevance. This section describes several papers on synthetic data application for UAV recognition systems.

"Dronesense" is the system for the identification, segmentation, and orientation detection of drones via neural networks [11]. CNN was employed with synthetic data generated using a generative adversarial network (GAN) and the Unreal Engine to create photorealistic images. The system was tested using the DJI MAVIC2 ZOOM and DJI INSPIRE2 drone models.

In [12], the authors proposed a UAV recognition system that utilizes a mixed dataset, comprising both real-world data and synthetic data generated by a deep convolutional generative adversarial network (DCGAN). The system was tested with 14 different drone models. The authors reported that their system achieved a high level of accuracy in recognizing UAVs from the images that were generated.

Publication [13] utilized an open-source 3D modeling program, Blender, to generate a synthetic dataset for the DJI Phantom, DJI Mavic, and DJI Inspire drone models. The dataset consisted of 1000 images rendered with 10 different backgrounds and textures, and a CNN was employed as a classification network using the fully synthetic dataset.

These studies demonstrate the potential of synthetic data in the design of UAV recognition systems, provide different synthetic data collection approaches, and highlight the advantages of synthetic data in overcoming the difficulties associated with obtaining large amounts of real-world data.

III. System Architecture

This research on real-world training data minimization for AI-based UAV recognition systems contains:

- classification neural network as a recognition system described in SectionIII-B.
- real-world dataset described in SectionIII-A1.
- synthetic data generator described in SectionIII-A2.

The experiment is conducted in two iterations:

- The initial iteration utilizes only real-world data to establish reference precision, recall, and f1-score values as a baseline.
- In the second iteration, a portion of the real-world data is replaced with synthetic data, with the proportion of synthetic data increasing by 10%. After that classification network is retrained with the new dataset, which allows comparison of both CNNs in case of their efficiency.

The subjects of the experiment are three drones: DJI Phantom 3, DJI Mavic Pro, and Parrot Bebop 2. The results of the experiment will provide insight into the effectiveness of using synthetic data in reducing the reliance on real-world data for UAV recognition systems, and the performance of the proposed system when applied to different drone models.

A. Training Dataset

This study uses two datasets, that represent real-world data and partly synthetic data. Datasets themselves represent a collection of 1000 images for each drone model, distributed as follows:

- 800 images as learning data;
- 100 images as validation data;
- 100 images as test data.

1) Real-world data: The real-world dataset was obtained through manual collection from open-access sources, followed by a manual filtering process. The images in the dataset exhibit variations in quality, drone-to-camera distance, drone position, environmental and weather conditions. After filtering, the dataset consisted of a total of 3000 images.

The primary limitation encountered during the acquisition of this data was the limited availability of images, resulting in a disproportionate distribution between high-quality and low-quality images. Images were captured in diverse environments, and they were taken at various angles and distances from the drone. This subsequently led to a non-uniformity in the dataset, affecting the performance of the recognition system.

2) Synthetic data: To create synthetic data the Stable diffusion neural network [14] was used, which, unlike the GAN [15], [16], allows to generate the necessary objects in various complex conditions, using text prompts or other images as reference. Stable Diffusion like other diffusion models [17] consists of 3 main parts: the variational autoencoder (VAE) [18], U-Net [19], and may contain an optional CLIP ViT-L/14 text encoder to condition the model on text prompts. The VAE encoder compresses the image from pixel space to a smaller dimensional latent space, capturing and learning a data distribution $p(x)$ main features and semantics of the image [14]. Gaussian noise is applied to the compressed latent representation during forward diffusion. The U-Net block denoises the output from forward diffusion backwards to obtain latent representation. At the last stage, VAE decoder generates the final image by converting the representation into pixel space (image). If model contains text-to-image transform, textual encoder translates text prompts to an embedding space. Stable diffusion was trained on images from the LAION-5B dataset which contains 5.85 billion CLIP-filtered image-text pairs, of which 2.32 billion contain English language. Considering those values, the neural network has an idea of what the object is called and looks like, allowing to create images according to a certain pattern:

1) photo or a painting;
2) subject of the photo - person, animal, object, or landscape;
3) extensions - lighting, color scheme, background, and other details;
4) specific art style or photo style.

For a new image generation, it was not enough to describe or give an image of a drone, because when specifying each object, a different, but similar object appears. Generating images of precisely the same drone was necessary to solve

this problem. To do this, the Stable diffusion extension, called “Dreambooth” [20], is used. This method takes a few images of a subject (in our case drone model) and the corresponding class name (e.g. “Mavic”) as input, and returns a fine-tuned/“personalized” text-to-image model that encodes a unique identifier that refers to the subject. Then, at inference, the unique identifier in different sentences to synthesize the subjects in different contexts can be implanted.

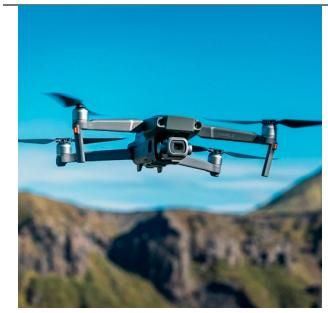
To train models for generating drones, datasets were prepared as follows: 15 images of each drone model in different angles and zoom, 512×512 pixels in size. This image size is also close to the one we will use for classification, which will allow us not to degrade the quality of the images. Stable diffusion training settings were as follows:

- ddim optimizer;
- 150 sample steps per image;
- text encoder training 50% of all training steps;
- Instance token is a drone model (Phantom 3, Mavic, Bebop) and class prompt UAV.

After training, the main parameter for creating synthetic images was writing text prompts to generate a drone in the needed conditions. Some examples of generated images and their comparison with real drone images can be seen in Table I.

The prompts can be divided into 2 groups: positive and negative prompts. Positive prompts define features that are desired to be present in the generated image. Negative prompts, on the other hand, define features to be excluded from the generated image. In this research, to generate drone pictures with different surroundings, the prompts have to be descriptive enough. But there is a problem that not all prompts from the description are taken into account in the generation process. This issue could lead to the generation of undesirable features that could potentially result in unsuccessfully generated image. Furthermore, some prompts are not understandable for the neural network, and, because of that, it will not apply these ”unknown” prompts for image generation. For example, such prompts as ”distant” or ”far away” were either ignored during the generation process or generated drones were severely deformed. Some unsuccessful drone generation results, as well as attempts to generate a picture of a distant drone, are shown in Fig. 1. To overcome the disregarding of specific prompts, the weighting of the prompts has to be performed. The weighting is done to increase the likelihood of the application of specific features for the generated image. This can be done by using brackets. The more brackets are applied to the specific prompt, the higher prompt’s priority during the generation process. This method of weighting is intuitive but it is not suitable for fine weighting of prompts. An alternative approach is to use numbers in prompts weighting. This provides the opportunity to fine-tune prompts priority. Besides prompts, there are some setups that affect the image generation result. The sampling method field defines the algorithm that will be applied during image generation. Output image dimensions are determined by width and height settings. The number of generated images

TABLE I. COMPARISON OF REAL DRONE PICTURES WITH GENERATED IMAGES

Real pictures of drones	Generated pictures of drones
	
	
	

can be controlled by setting the batch size and batch count. Adjusting the classifier-free guidance (CFG) scale, in its turn, changes the fidelity between the prompt and output images.

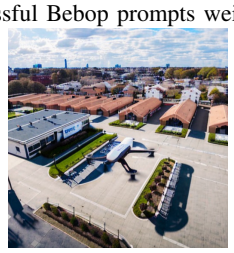
After prompts writing, prompts weighting, and Dreambooth optimal setup, all three drone models (DJI Mavic, Parrot Bebob 2, and DJI Phantom 3) were generated in different surroundings, especially cities, deserts, and island-type surroundings, as well as in some weather conditions when it could be difficult to capture flying drone, for example, rain and snow. All Dreambooth settings for drone image generation are summarized in Table II.

TABLE II. DREAMBOOTH SETTINGS

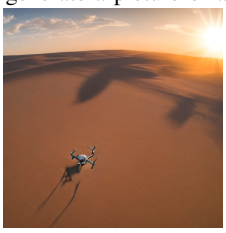
Dreambooth setting	**Value or applied method**
Sampling method	Euler a
Sampling steps	80-100
Width	512
Height	512
CFG scale	8-10

Examples of drone generation prompts and their results are shown in Table III.

(a) Unsuccessful Bebop prompts weighting result



(b) Attempt to generate a picture of a distant Bebop



(c) Attempt to generate a picture of a distant Mavic

Fig. 1. Unsuccessful drone generation results.

B. Classification Network

In this section, we will look at the process of training and testing a neural network for recognizing drone models. The performance and accuracy of the model for CNN training cases 1) on real data only, described in Section III-A1, and 2) with the addition of synthetic data, presented in Section ere compared. Since the main objective of this research was to classify the drone in real life, only real photos were used for testing. All datasets contained 1000 images of each drone model. The images were split into 3 groups: 80% training, 10% validation, and 10% testing.

The neural network was implemented using the Tensorflow and Keras libraries in the Python programming language. The training and testing process was performed on a high performance computer (HPC) with a Nvidia A100 graphic card. As CNN we choose EfficientNet-B5 checkpoint for transfer-learning [21], [22], which is pre-trained on the ImageNet [7] dataset. EfficientNets are a family of image classification models, which achieve state-of-the-art accuracy, yet being an order-of-magnitude smaller and faster than previous models [23]. Of course, at the moment of writing this article, improved versions of this neural network [24] exists. However, mentioned version with default settings is also capable of classifying with transfer learning state-of-the-art accuracy on transfer learning datasets till 98.8% [23], what is enough for our task. This model is designed to process images with sizes up to 456×456 pixels, which is quite enough for classifying our synthetic data with a size of 512×512 without much quality loss. All used CNN hyperparameters are shown in Table IV.

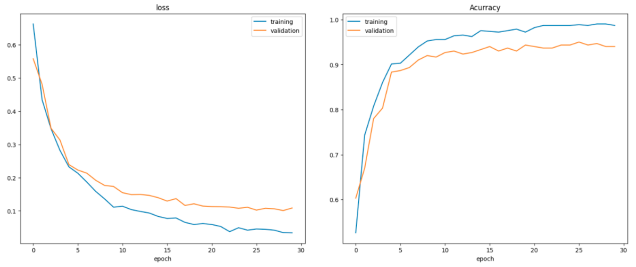
TABLE III. EXAMPLES OF DRONE GENERATION PROMPTS

Positive prompts: A photo of flying quadrocopter bebop : 7, city at the background : 3, buildings : 2, top of buildings : 1 **Negative prompts**: POV : 7, bad geometry: 2, close-up : 1, bad proportions: 3, distorted perspective : 4	
Positive prompts: A photo of flying quadrocopter bebop : 7, dunes at the background : 4, desert: 1 **Negative prompts**: POV : 7, bad geometry: 2, close-up : 1, bad proportions: 3, distorted perspective : 4	
Positive prompts: A photo of flying quadrocopter bebop : 5, island at the background : 4, dream archipelago: 2 **Negative prompts**: 7, bad geometry: 2, close-up : 1, bad proportions: 3, distorted perspective : 4	

TABLE IV. CNN HYPERPARAMETERS

Neural network hyperparameters	
Batch size	16
Epoch	30
Learning rate	min=0.00001, max=0.00005
Dropout	0.4
Optimization	Adam

Training process results are shown on the image 2, where we can see how the Accuracy and Loss function depends on epoch. Evaluating the resulting graphs, we can see that the results of the training and validation accuracy after the 20th epoch do not change much, and the graph has reached a plateau. On the loss plot, the results hardly change after 17 epochs. After evaluating this information and in order to avoid over-fitting, it was decided to stop at the 15th epoch. The results of the model trained for 15 epochs are shown in



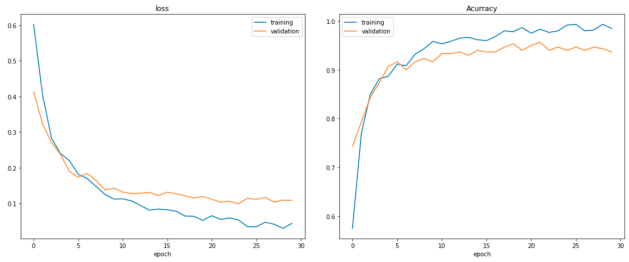
(a) Dependence of the loss values on the epoch

(b) Dependence of the accuracy values on the epoch

Fig. 2. First CNN results training 30 epoch.

the TableV.

In the second experiment, the same CNN, where 10% files in the training data set were replaced with images generated by the Stable diffusion neural network, were trained. Taking into account the peculiarities of generating images from Section III-A2, only close-range drone photos were replaced with close-range synthetic images. The second CNN training history is shown in Fig. 3, at the 15th epoch the losses increased and the accuracy decreased, but finally the graph flattens out. The 15th epoch has been selected for the test with both neural networks. The results of the second model trained for 15 epochs are shown in the Table VI.



(a) Dependence of the loss values on the epoch

(b) Dependence of the accuracy values on the epoch

Fig. 3. Second CNN results training 30 epoch.

IV. Results

To assess the performance of the classification, the confusion matrix [25], where are calculated such parameters as accuracy, precision, recall, and f1-score on a per-class basis, was employed. The metrics are calculated using true and false positives, true and false negatives. Positive and negative in this case are generic names for the predicted classes. Confusion matrix results are interpreted in the classification report, shown in Table V. Total accuracy of the first model is 95.33%. The classification report of testing the second neural network, which contains the synthetic images, is in Table VI. This model's total accuracy is 96.0%

Comparing the accuracy results of both neural networks, we can see that the neural network which was trained on a mixed dataset works better for 0.67%. This small difference

TABLE V. CLASSIFICATION REPORT FOR 15 EPOCH WITH REAL DATA

	precision	recall	f1-score	support
Mavic	0.95	0.98	0.97	100
Bebop	0.99	0.90	0.94	100
Phantom	0.92	0.98	0.95	100
accuracy			0.95	300
macro avg	0.95	0.95	0.95	300
weighted avg	0.95	0.95	0.95	300

TABLE VI. CLASSIFICATION REPORT FOR 15 EPOCH WITH SYNTHETIC DATA

	precision	recall	f1-score	support
Mavic	0.92	0.97	0.95	100
Bebop	0.99	0.96	0.97	100
Phantom	0.97	0.95	0.96	100
accuracy			0.96	300
macro avg	0.96	0.96	0.96	300
weighted avg	0.96	0.96	0.96	300

may be due to the fact that on the synthetic images drones are in very different situations and we can think of it as data augmentation [26], [27] - we have diversified the dataset. The most common drone photos in the real dataset are bottom-up photos of the drone when it flies in the sky. Of course, there were other photos taken from other angles and different places, but on synthetic images we generated very rare situations. For example, a drone in the desert from a Table III.

V. Discusssion

This study aimed to investigate the potential of synthetic data in the design of UAV recognition systems. Our results showed that by replacing a portion of the real drone images with synthetic ones in the training dataset, comparable accuracy in the performance of the CNN, which has been trained with real-world data only, can be obtained.

One issue that arose during the study was the imperfection of the real-world dataset. The data was collected from open-access sources and was not perfectly matched in terms of parameters such as high/low-quality images count proportion. This imbalance in the quantity of high and low-resolution images for each UAV model can potentially affect the performance of the CNN in recognizing and classifying the different drone models [28].

Also, it should be noted that this study used the default version of Stable diffusion and Dreambooth extension without any extra VAE [29], which could lead to some limitations on photo-realism. For example, Dreambooth drones generation distance is limited by the proximity of the "camera". This can result in drone geometry imperfections and deformations that could lead to unsuccessful distant drone generation. That was one of the reasons that limited dataset diversity and size. However, this limitation can potentially be mitigated through the secondary generation of background images that artificially distance the UAVs.

However, this is an ongoing research area and future studies will focus on improving the photo-realism of synthetic images

of the UAVs and increasing the proportion of the synthetic data in the training dataset. And, despite all of these limitations, the results of this study demonstrate the potential of synthetic data in overcoming the difficulties associated with obtaining large amounts of real-world data. The use of synthetic data allows the generation of large and diverse datasets with a high level of control over the parameters and scenarios. That could potentially accelerate the development of new UAV recognition systems.

VI. Competing Interests

Authors declare no competing interests.

References

[1] V. Matić, V. Kosjer, A. Lebl, B. Pavić, and J. Radivojević, "Methods for drone detection and jamming," in *Proceedings of the 10th International Conference on Information Society and Technology (ICIST)*, 2020, pp. 16–21.

[2] "Modern methods for UAV detection, classification, and tracking." IEEE, oct 2022, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9978860/

[3] F. Svanström, C. Englund, and F. Alonso-Fernandez, "Real-time drone detection and tracking with visible, thermal and acoustic sensors," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 7265–7272.

[4] J. Klare, O. Biallawons, and D. Cerutti-Maori, "Uav detection with mimo radar," in *2017 18th International Radar Symposium (IRS)*, 2017, pp. 1–8.

[5] A. Martian, F.-L. Chiper, R. Craciunescu, C. Vladeanu, O. Fratu, and I. Marghescu, "Rf based uav detection and defense systems: Survey and a novel solution," in *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 2021, pp. 1–4.

[6] L. Rauh, S. Gärtner, D. Brandt, M. Oberle, D. Stock, and T. Bauernhansl, "Towards ai lifecycle management in manufacturing using the asset administration shell (aas)," *Procedia CIRP*, vol. 107, pp. 576–581, 2022, leading manufacturing systems transformation – Proceedings of the 55th CIRP Conference on Manufacturing Systems 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827122003122

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[8] S. Dodge and L. Karam, "A study and comparison of human and deep learning recognition performance under visual distortions," 2017. [Online]. Available: https://arxiv.org/abs/1705.02498

[9] J. S. Ross and H. M. Krumholz, "Ushering in a New Era of Open Science Through Data Sharing: The Wall Must Come Down," *JAMA*, vol. 309, no. 13, pp. 1355–1356, 04 2013. [Online]. Available: https://doi.org/10.1001/jama.2013.1299

[10] D. Duplevska, M. Ivanovs, J. Arents, and R. Kadiķis, "Sim2real image translation to improve a synthetic dataset for a bin picking task," 09 2022, pp. 1–7.

[11] S. Scholes, A. Ruget, G. Mora-Martín, F. Zhu, I. Gyongy, and J. Leach, "Dronesense: The identification, segmentation, and orientation detection of drones via neural networks," *IEEE Access*, vol. 10, pp. 38 154–38 164, 2022.

[12] C. Li, S. C. Sun, Z. Wei, A. Tsourdos, and W. Guo, "Scarce data driven deep learning of drones via generalized data distribution space," 2021. [Online]. Available: https://arxiv.org/abs/2108.08244

[13] M. Wisniewski, Z. A. Rana, and I. Petrunin, "Drone model classification using convolutional neural network trained on synthetic data," *Journal of Imaging*, vol. 8, no. 8, 2022. [Online]. Available: https://www.mdpi.com/2313-433X/8/8/218

[14] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2021. [Online]. Available: https://arxiv.org/abs/2112.10752

[15] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," 2018. [Online]. Available: https://arxiv.org/abs/1809.11096

[16] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2105.05233

[17] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," 2015. [Online]. Available: https://arxiv.org/abs/1503.03585

[18] L. Cinelli, M. Marins, E. da Silva, and S. Netto, *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer International Publishing, 2021. [Online]. Available: https://books.google.lv/books?id=N5EtEAAAQBAJ

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[20] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," 2022.

[21] Callidior, "Keras-applications: Reference implementations of popular deep learning models." https://github.com/Callidior/keras-applications/, accessed on 25.01.2023.

[22] Tensorflow, "Efficientnet for tensorflow/tpu," https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet, accessed on 25.01.2023.

[23] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1905.11946

[24] ——, "Efficientnetv2: Smaller models and faster training," 2021. [Online]. Available: https://arxiv.org/abs/2104.00298

[25] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425797000837

[26] A. Patrizi, G. Gambosi, and F. M. Zanzotto, "Data augmentation using background replacement for automated sorting of littered waste," *Journal of Imaging*, vol. 7, no. 8, 2021. [Online]. Available: https://www.mdpi.com/2313-433X/7/8/144

[27] T. Xie, X. Cheng, X. Wang, M. Liu, J. Deng, T. Zhou, and M. Liu, "Cut-thumbnail: A novel data augmentation for convolutional neural network," in *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, oct 2021. [Online]. Available: https://doi.org/10.1145%2F3474085.3475302

[28] M. Koziarski and B. Cyganek, "Impact of low resolution on image recognition with deep neural networks: An experimental study," *International Journal of Applied Mathematics and Computer Science*, vol. 28, pp. 735–744, 12 2018.

[29] K. Pandey, A. Mukherjee, P. Rai, and A. Kumar, "Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents," 2022. [Online]. Available: https://arxiv.org/abs/2201.00308