Pattern Recognition Letters 193 (2025) 101–107

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Enhanced generation of automatically labelled image segmentation datasets by advanced style interpreter deep architectures

Marcos Sergio Pacheco dos Santos Lima Junior [a,*], Ezequiel López-Rubio [a,b], Juan Miguel Ortiz-de-Lazcano-Lobato [a,b], José David Fernández-Rodríguez [a,b]

[a] Department of Computer Languages and Computer Science. University of Málaga, Bulevar Louis Pasteur, 35, Málaga, 29071, Spain
[b] Biomedical Research Institute of Málaga (IBIMA), C/ Doctor Miguel Díaz Recio, 28, Málaga, 29010, Spain

ARTICLE INFO

Editor: Jiwen Lu

Keywords:
Image segmentation
Convolutional neural networks

ABSTRACT

Large image datasets with annotated pixel-level semantics are necessary to train and evaluate supervised deep-learning models. These datasets are very expensive in terms of the human effort required to build them. Still, recent developments such as DatasetGAN open the possibility of leveraging generative systems to automatically synthesise massive amounts of images along with pixel-level information. This work analyses DatasetGAN and proposes a novel architecture that utilises the semantic information of neighbouring pixels to achieve significantly better performance. Additionally, the overfitting observed in the original architecture is thoroughly investigated, and modifications are proposed to mitigate it. Furthermore, the implementation has been redesigned to greatly reduce the memory requirements of DatasetGAN, and a comprehensive study of the impact of the number of classes in the segmentation task is presented.

1. Introduction

Deep learning models for image analysis or synthesis often require massive amounts of data to be trained. When training datasets lack sufficient data, models are prone to overfitting, which significantly reduces their generalisation capability. Certain demanding tasks, such as biomedical image processing, require robustness against particular types of data noise or distortions to achieve an acceptable level of accuracy [1]. In those cases, annotating hundreds or even thousands of data images at the pixel level is essential for deep neural networks to perform adequately. This process is especially burdensome considering that supervised and semi-supervised deep learning models for image processing require semantic information of various kinds, such as bounding boxes, object or scene-level key points, depth fields or segmentation masks. More specifically, image segmentation is often challenging for tasks with limited datasets or fine-grained annotations [2]. While significant efforts have been devoted to generating annotated images by rendering 3D models [3], even leveraging deep learning to generate image annotations [4], this work focuses on the employment of deep learning for the joint synthesis of realistic images and their corresponding semantics.

Recently, both Generative Adversarial Networks (GANs) and Diffusion Models have been utilised to synthesise annotated images [5,6]. Diffusion models are the latest generative paradigm and often achieve better generalisation. However, they are significantly more computationally expensive than GANs. In addition, controlling the latent space of diffusion models remains challenging and less understood, making them a more complex alternative than GANs for tasks such as image editing and projection [6–9]. Therefore, research related to methods that enhance state-of-the-art GAN-based approaches still remains highly relevant.

A key concept of GANs is their ability to synthesise realistic images by mapping vectors from a *latent space*, which encodes complex semantic information. This characteristic makes GANs valuable tools for domain-specific image processing tasks with intricate semantic requirements, such as generating portraits from sketches [10], text-to-fashion image synthesis [11], or anomaly detection in small video snippets [12].

There have been various efforts to adapt GANs for the joint generation of realistic images and their semantics. One general strategy to achieve this is to explicitly use semantic information to modulate the synthesis of the images. For example, the vectors from the latent space can be manipulated and characterised to enable the generation of images with specific semantic constraints [13]. A related approach is to modify the GAN's underlying deep learning model such that the semantic information is used to configure the synthesised image [14].

* Corresponding author.
E-mail addresses: marcos.pacheco@uma.es (M.S. Pacheco dos Santos Lima Junior), ezeqlr@lcc.uma.es (E. López-Rubio), jmortiz@lcc.uma.es (J.M. Ortiz-de-Lazcano-Lobato), josedavid@lcc.uma.es (J.D. Fernández-Rodríguez).

https://doi.org/10.1016/j.patrec.2025.04.021
Received 2 November 2022; Received in revised form 25 February 2025; Accepted 14 April 2025
Available online 23 April 2025

Another strategy uses GANs to synthesise images and annotations by leveraging intermediate tensors from the model's inner layers to encode semantic properties of the generated images. This information can be extracted in multiple ways. For example, semantics can be obtained by unsupervised clustering of the intermediate tensors of the GAN [15]. In this context, a prominent strategy was proposed by DatasetGAN [16], using StyleGAN [17] intermediate tensors to train an ensemble of multilayer perceptrons (MLPs) that predicts semantic information per pixel. Due to its architecture, DatasetGAN requires only a few labelled examples to generalise across the latent space, which opens the possibility of generating extensive datasets comparable in scale and number of image classes to ImageNet [18]. Consequently, it has attracted significant research interest and has been applied in various domains, some examples are augmenting medical images [19,20], generating annotated images of tunnel cracks [21] and trucks [22].

Following the publication of StyleGAN, several efforts aimed to improve image quality, with a significant breakthrough achieved by StyleGAN2-ADA [23]. For this reason, there have been initiatives to replace DatasetGAN's backbone with StyleGAN2-ADA, although, solely, it does not directly impact how DatasetGAN's style interpreter interacts with its feature map [19,22,24]. Likewise, other studies focus on equivalent alternatives, but use diffusion models as a backbone [25,26].

On a different direction, BigdatasetGAN [18] focused on enabling multi-class image synthesis. By manually annotating a few images per class on ImageNet and training an image segmentation architecture on top of BigGAN instead of StyleGAN, the study allowed the creation of large-scale annotated datasets.

Still, another related model with a DatasetGAN-derived framework that stood out is HandsOff [27]. By unifying GAN inversion with dataset generation, it enables the synthesis of datasets with rich pixel-wise labels in multiple scenarios without the need to manually annotate synthetic images.

Even though several improvements have been achieved, we observed that in DatasetGAN and many of its successors employing StyleGAN-based backbones, the connection with feature maps consists of simple MLPs, a design choice that has neither been demonstrated to be particularly distinctive nor received significant attention in prior works. Furthermore, the impact of the aforementioned improvements cannot be directly compared to changes in the style interpreter architecture solely, which could be addressed as a separate component in the overall framework. Still, the DatasetGAN publication does not detail the quantity of annotated masks necessary to avoid underfitting or overfitting. To be more specific, in our experiments, we have found that it is prone to overfitting.

This work addresses these issues by exploring the DatasetGAN paradigm, where our main contributions are:

- The proposition of a novel architecture that uses neighbouring pixel information and an innovative data loader to improve the quality of image segmentation.
- An extensive analysis of the overfitting behaviour in the original architecture and the presentation of modifications to mitigate it.
- An improved computational strategy, which drastically reduces computing requirements.
- A comprehensive study of the impact of the number of classes on DatasetGAN's overall performance.

Our code is available at https://github.com/icai-uma/datasetGAN2_release and the following sections are structured as follows: Section 2 presents the proposed methodology, Section 3 reports the experiments supporting our proposal, and Section 4 explains our conclusions.

2. Methodology

In order to properly evaluate modifications to the DatasetGAN architecture, we formulated a strategy to compare them with its original proposition.

Table 1
Comparison per dataset of the RAM necessary to run the original DatasetGAN method versus our 4-fold cross-validation strategy.

	Face	Cat	Bedroom
Original method	125.8 GB	61.2 GB	49.2 GB
Our method	4.3 GB	4.3 GB	4.3 GB

2.1. Comparison strategy

The strategy employed in this work uses k-fold cross-validation with $k = 4$ and M images in each fold to compare different MLP pixel classifier architectures. Each fold comprises two networks with different seeds to account for the effect of the consensus decision of the ensemble of classifiers presented by DatasetGAN authors, where each network is referred to as CV_i, and $1 \leq i \leq 8$ is the network index.

In this process, a validation loop is carried out at the end of every training epoch. The metrics are then stored so that the mean and individual cross-validation scores might be computed. Fig. 1 shows an overview of this strategy for a dataset containing images with d_1 width, d_2 height, and d_3 features split along the 18 styles blocks defined by StyleGAN.

2.2. Improvements in DatasetGAN architecture

The main focus of this work is to study ways of improving DatasetGAN by changing the architecture of the pixel classifiers inside its style interpreter. In this regard, one of our propositions is to add dropout layers to attenuate overfitting. Given that the dropout probability p is the probability that a single neuron is kept during each training pass [28], we convention in our experiments q as the complementary probability of the neuron being dropped out. When added after Batch Normalisation layers, dropout layers are known to provide a more stable training process, faster convergence speed, and better accuracy results. It is possible because the inputs of subsequent layers can be whitened, as the correlation is linearly reduced and the mutual information between neurons is quadratically reduced with respect to the dropout probability (p) [29].

Then, we propose to embed these dropout elements on a new architecture, where convolutions are employed to extract information from neighbouring pixels to benefit image segmentation. This proposition is detailed in Section 3.

2.3. Computation strategy

We also propose a modification in the style interpreter training computation. The original method creates the feature map used for training the style interpreter and stores it in the Random Access Memory (RAM). Then, it shuffles all the training pixels within the feature map and trains the MLP. We observed through experimentation that this strategy might require hundreds of GB of RAM, depending on the dataset.

Our strategy differs by creating and storing the feature maps on the hard disk. Then, the training is performed by loading the training data in chunks into the RAM in random order. The validation loop described in Section 2.1 is executed accordingly, with the feature maps corresponding to the validation images loaded in chunks into the RAM. A comparison between the strategies is shown in Table 1.

Additionally, in the original DatasetGAN strategy, the feature map is recomputed every time the interpreter is trained. The feature map is identical for every execution if the DatasetGAN latent codes and the StyleGAN checkpoint are the same. Given that we tested several different configurations of the MLP pixel classifier, keeping the feature map on the hard disk was also convenient to avoid spending time on redundant executions.

Furthermore, we employ a custom data loading strategy for training the architecture proposed in Section 3.1.2, which is detailed in the same section. This approach allows loading pixels in random order along with their neighbouring information and assures training generalisation combined with efficient memory management.

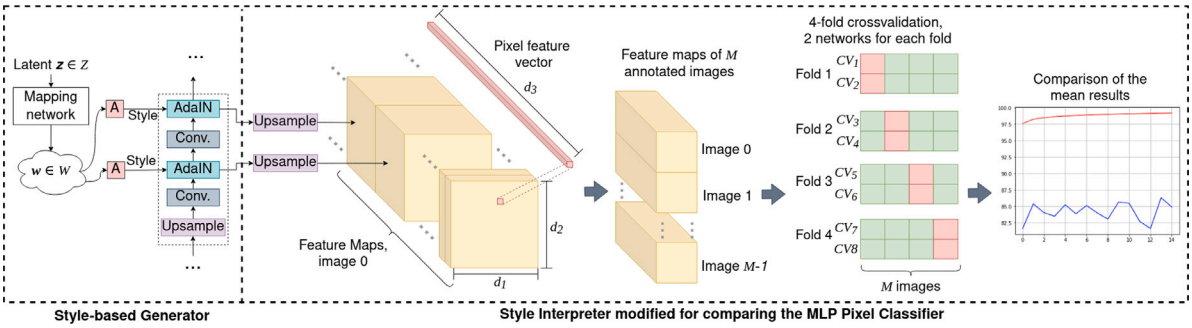

Fig. 1. Overview of the comparison strategy adopted containing a modified style interpreter.

3. Experiments

We analysed three datasets with a different number of classes C: Face ($C = 34$), Cat ($C = 16$), and Bedroom ($C = 10$). The Cat and Face datasets are the same used by the DatasetGAN authors [16], while the Bedroom dataset joins the checkpoint provided by the StyleGAN authors [17] with training annotations made by our research group.

As this study employs a 4-fold crossvalidation strategy, it was convenient for each dataset to contain a multiple of 4 images. Accordingly, all 16 available annotated images were used for the Face dataset, 28 of the 30 available annotated images were selected for the Cat dataset, and 24 images were annotated for the Bedroom dataset.

In the experiments, all the training loops were extended from 4 epochs in the code provided by the DatasetGAN authors to 15 epochs (named from epochs 0 to 14). It allowed better visualisation of the networks' stabilisation.

3.1. Quantitative results

To support our proposed architecture, different configurations were investigated for each of the datasets. As we observed that the original architecture is prone to overfitting, it was convenient to start by finding architectural changes to mitigate it.

3.1.1. Evaluation of overfitting in DatasetGAN

We evaluate the overfitting reduction in DatasetGAN by adding dropout layers after each of the batch normalisation layers of the architecture. For simplicity, we name this new architecture shown in Fig. 2(b) as the Dropout architecture, where the use of combinations of dropout probabilities q from a set S of values is investigated. In this experiment, $S = \{q = 20\%, q = 40\%, q = 60\%, q = 80\%\}$, and q_1 and q_2 were assigned as the dropout probabilities of the first and second layers, respectively. For each of the combinations of $q_1, q_2 \in S$, the training and validation accuracy and loss were plotted in Fig. 3 for the Face dataset, Fig. 4 for the Cat dataset, and Fig. 5 for the Bedroom dataset.

Some interestingly similar characteristics might be noticed from these plots in all these experiments. First, when looking at the original DatasetGAN MLP pixel classifier individually in all of the plots, the training accuracy values are very high, while the training loss values are low. On the contrary, the validation accuracy values are not as high as the training ones, and the validation loss values reach extremely large values. These observations strongly suggest that the MLP pixel classifier from the original DatasetGAN is overfitting for all three datasets. This issue can be ameliorated by using Dropout layers.

Secondly, the experiments with dropout layers consistently improved the validation accuracy for most dropout probability combinations. For instance, in epoch 14, all the dropout combinations outperformed the original architecture in the Face and Bedroom datasets. Similarly, in the same epoch, 12 out of the 16 dropout probability combinations were superior to the original approach in the case of the Cat dataset.

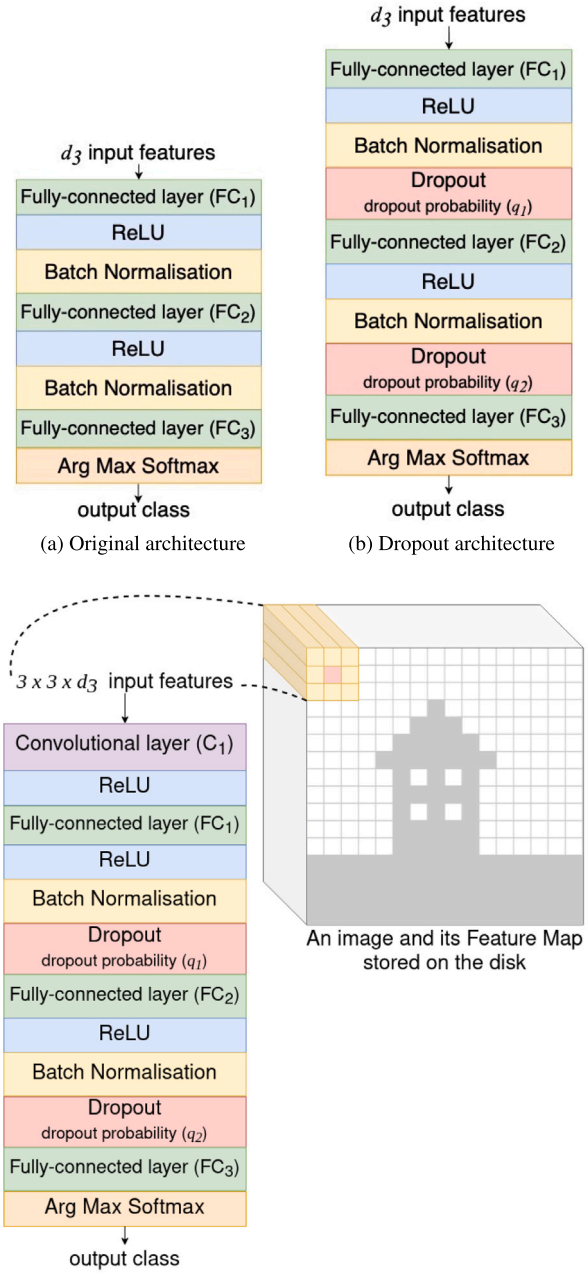

Fig. 2. Proposed modifications in DatasetGAN architecture: (a) Original architecture; (b) Dropout architecture; and (c) Proposed architecture.

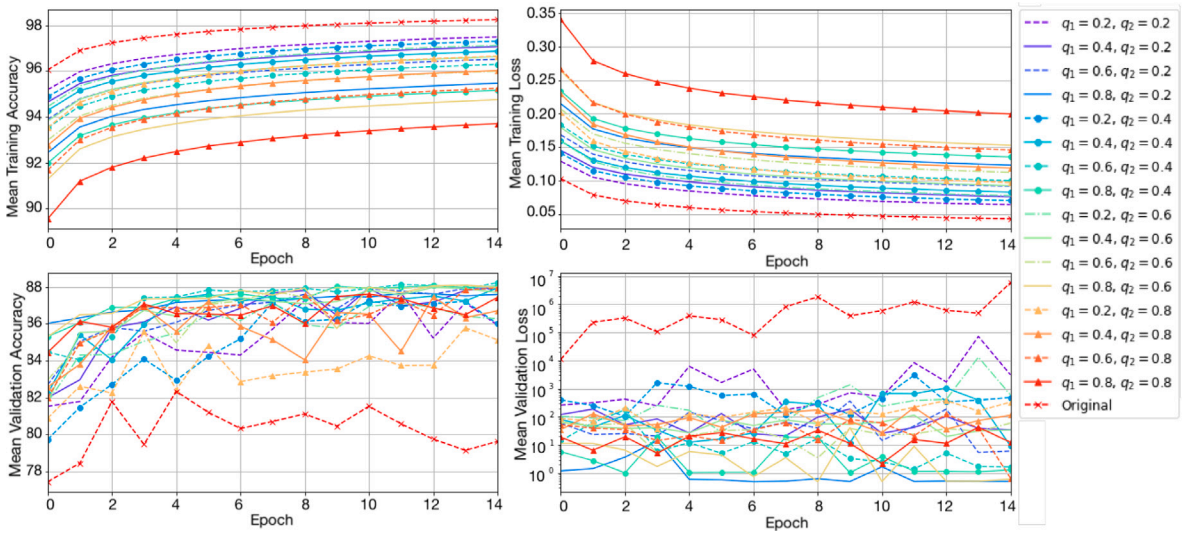

Fig. 3. Effect of the variation of the dropout probabilities q_1 and q_2 for the Face dataset: mean training accuracy at the upper left, mean training loss at the upper right, mean validation accuracy at the lower left, and mean validation loss at the lower right.

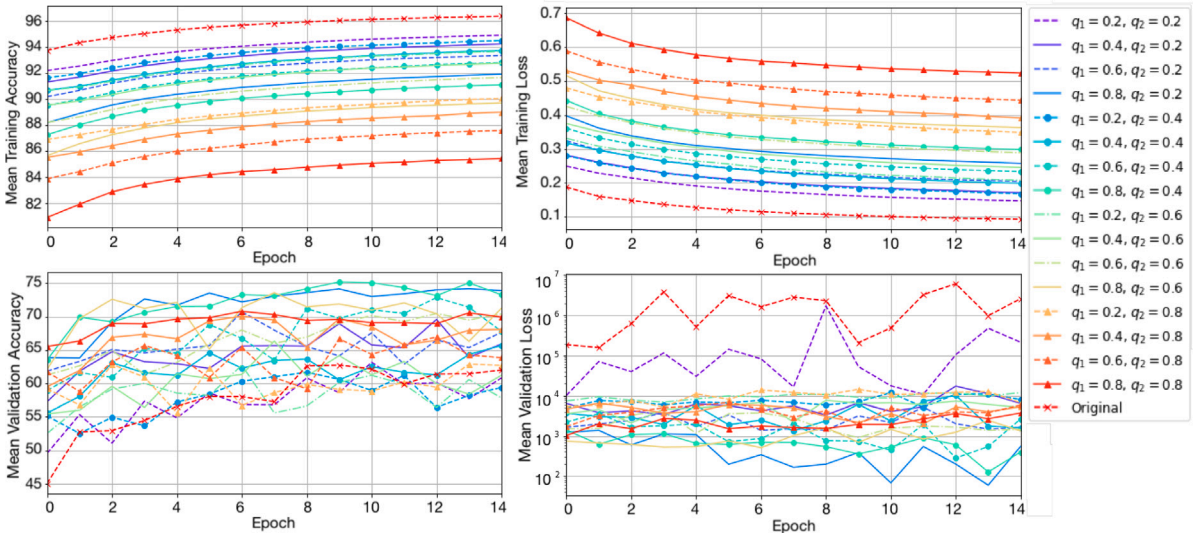

Fig. 4. Effect of the variation of the dropout probabilities q_1 and q_2 for the Cat dataset: mean training accuracy at the upper left, mean training loss at the upper right, mean validation accuracy at the lower left, and mean validation loss at the lower right.

Additionally, for all the validation accuracy plots, the combinations with higher values of q_1, in general, performed better. One possible explanation is that the number of features in the feature maps produced by the Style-Based Generator (d_3) is more than what is necessary to segment the images correctly. The feature map contains 4992 features for both the Cat and Bedroom datasets and 5088 features for the Face dataset. The total number of features in the feature map depends on the image resolution and other hyperparameters [17].

It can also be observed that the best validation accuracy values do not coincide in the same combinations of q_1 and q_2 for the three datasets. Instead, there is a different set of best combinations for each dataset, as shown in Table 2, with q_1 between 60%–80% in all cases and q_2 between 20%–40% in all cases.

3.1.2. Proposed architecture

Another hypothesis studied is whether the pixel classifier would provide better segmentation quality if being fed with information about its neighbouring pixels or not. To evaluate it, we created a custom data loader that returns a $b_{size}\times k_s\times k_s\times d_3$ matrix from the feature map during training iterations, where b_{size} is the batch size, $k_s = 3$, and $k_s \times k_s$ is the kernel size of a convolutional input layer. This way, it returns pixels along with their features picked from the centre of a $k_s \times k_s$ window containing information of adjacent pixels.

The custom data loader proposed picks the training data randomly instead of using a sliding window, which assures the generalisation in the training process as pixels close to each other are expected to have similar patterns. It is accomplished without shuffling the entire dataset at once, which would be costly in terms of RAM and would not allow retrieving adjacent pixels. Instead, for each epoch, we map the positions of the pixels not yet loaded, then pick batches of positions randomly, and load them along with their neighbouring pixels.

The neural network architecture used in this process has a convolutional input layer with N_k kernels, followed by two blocks, each containing a fully connected layer, a ReLU activation, a batch normalisation, and a dropout layer. Then, the last block is connected to another fully connected layer with softmax. The values of the dropout probabilities q_1 and q_2 presented in the scheme of this architecture in Fig. 2(c) were set to 60% and 40%, respectively. The choice of these values was based on the good performance that this combination

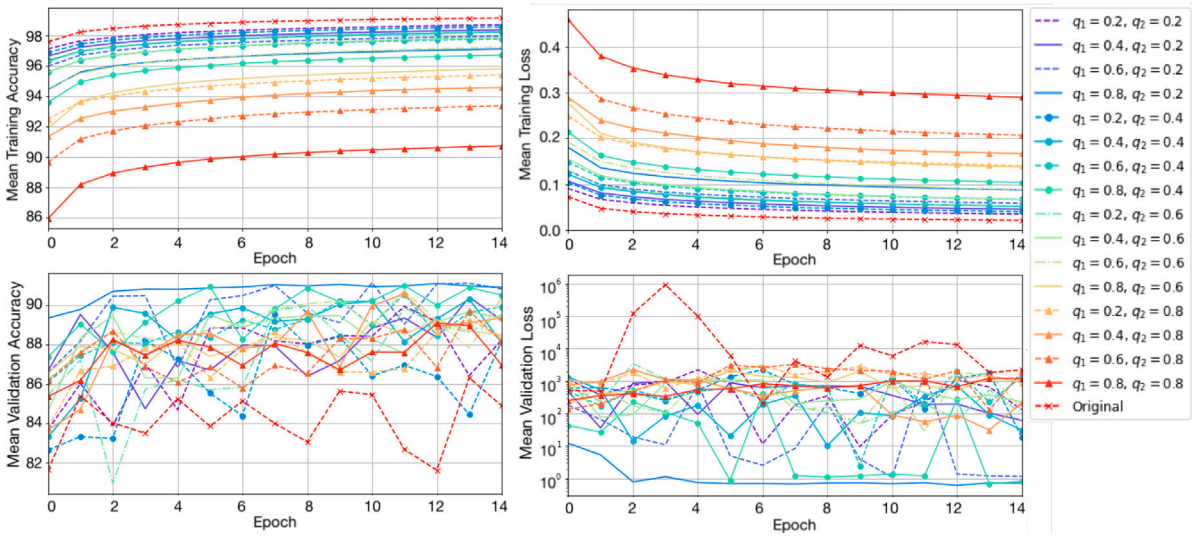

Fig. 5. Effect of the variation of the dropout probabilities q_1 and q_2 for the Bedroom dataset: mean training accuracy at the upper left, mean training loss at the upper right, mean validation accuracy at the lower left, and mean validation loss at the lower right.

Table 2
Comparison of the performance of DatasetGAN with the best combination of q_1 and q_2 in the Dropout architecture, and with the Proposed architecture.

Architecture	Face		Cat		Bedroom	
	% Validation accuracy	% (q_1, q_2)	% Validation accuracy	% (q_1, q_2)	% Validation accuracy	% (q_1, q_2)
DatasetGAN	82.341	–	62.740	–	86.311	–
+ dropout layers (Dropout architecture)	88.218	(60, 40)	75.178	(80, 40)	91.114	(60, 20)
Proposed architecture	**88.567**	(60, 40)	**77.903**	(60, 40)	**91.300**	(60, 40)

achieved for the all datasets in Section 3.1.1. Additionally, we set $N_k = 2 \times d_3$ by convention in our experiments.

The comparison of the results obtained with our architecture against DatasetGAN is shown in Table 2, where it is possible to observe a significant performance improvement for all the datasets evaluated.

Although the dropout layers play an important role in reducing overfitting and improving overall performance, we observe that even the best configuration found with the dropout architecture is overtaken by the proposed architecture. These results support the hypothesis that the pixel classifier performs better when inputted with information about its neighbouring pixels.

3.1.3. Impact of the number of classes in the segmentation task

Following, it was investigated if the quantity of classes intended for the semantic segmentation task impacts its performance. The Face dataset was chosen for this task because it has the highest number of classes. Then, the original architecture of DatasetGAN was used to train a simpler version of the Face dataset. In this version, the original 34 classes were regrouped into 10 and more simple classes (background, hair, neck, mouth, nose, eyes, forehead, ears, eyebrows/moustache, and the remaining area of the face). After that, another experiment was done with a stronger simplification, which regrouped the 34 classes into 2: background and head.

These experiments followed the same 4-fold cross-validation methodology presented in Section 2.1, and its results are displayed in Fig. 6. It is noticeable that as the number of classes decreases, the performance of the pixel classifiers in the segmentation task significantly increases. This behaviour suggests that DatasetGAN is more robust when targeting fewer classes.

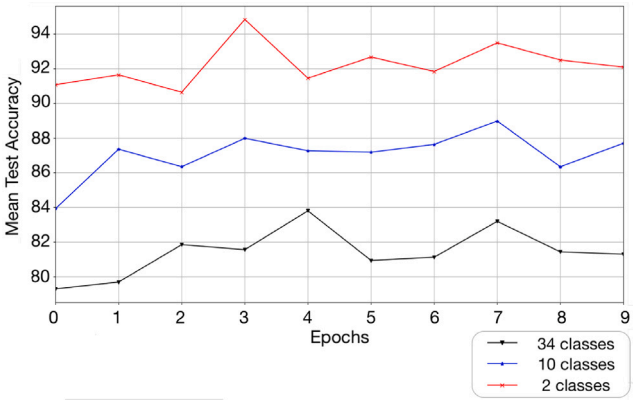

Fig. 6. Mean validation accuracy achieved for the Face dataset considering different number of classes in the segmentation task.

3.2. Qualitative results

It was also convenient to compare the cross-validation results presented in Section 3.1 qualitatively. Thus, it is displayed in Fig. 7 a few of the segmentation results generated by single networks during the cross-validation described in Section 2.1. In this figure, we compared the ground truth annotations (Fig. 7a) with single network results of the original architecture (Fig. 7b) and single network results of the proposed architecture (Fig. 7c). It was also shown the difference between networks within the same fold for the original architecture (Fig. 7d) and the proposed architecture (Fig. 7e). Further, it was computed the difference between single network results and the ground truth for the

Fig. 7. Examples of the qualitative improvements observed in the Face, Cat, and Bedroom datasets: (a) ground truth segmentation; (b) segmentation resulted from the original DatasetGAN architecture; (c) segmentation resulted from the proposed architecture; (d) difference between the two cross-validation networks within the same fold for the original DatasetGAN and the proposed architectures; (e) difference between the two cross-validation networks within the same fold for the proposed architecture; (f) difference between the ground truth segmentation and the original DatasetGAN architecture; (g) difference between the ground truth segmentation and the proposed architecture.

original architecture (Fig. 7f), and for the proposed architecture (Fig. 7g).

Comparing the Fig. 7a, b, and c, reveals that the proposed architecture achieves a noticeable improvement. It is important to note that the segmentation imperfections perceptible in these figures tend to improve if the consensus decision of network ensembles are visualised [30]. Fig. 7d and e confirm it, where some wrong segmentation artefacts are not present in the network pair of the same fold.

Additionally, it is apparent that the original pixel classifier networks fail more often to agree on the same segmentation class (the green areas show the divergence in the class choice of the networks within the same fold). Finally, it is possible to notice from Fig. 7f and g that both strategies fail consistently within the borders of the classes.

4. Conclusions

Generative systems based on deep learning (such as GANs) can synthesise semantically coherent images. In this context, DatasetGAN might be understood as a pioneering effort to extract (and render in usable form) semantic information from such generative systems. Due to the significance of DatasetGAN, a lot of research interest has been directed towards it. However, as a work of seminal nature, it provides an initial demonstration of a new concept rather than a polished one, where DatasetGAN's seminal model can be improved in various ways. Some of the shortcomings we addressed are the original design of the style interpreter, a 3-layer perceptron prone to significant overfitting, and an oversimplified computational strategy, limiting its suitability for real-world applications.

In this work, the DatasetGAN architecture was analysed in detail, and an exhaustive study on overfitting mitigation was presented. Firstly, dropout layers were introduced as a regularisation mechanism against overfitting after batch normalisation layers, substantially improving performance. Then, we employed convolutions to benefit from semantic information from neighbouring pixels via a custom data loader that assured a consistent generalisation during the training process. As a result, we proposed a new architecture that achieved significantly enhanced image segmentation quality compared to the original model. Furthermore, the training process was optimised to reduce RAM requirements, enabling model training and inference on a less expensive hardware. Finally, we evaluated the impact of the number of classes in the segmentation task, finding that scenarios with fewer classes achieved better performance.

CRediT authorship contribution statement

Marcos Sergio Pacheco dos Santos Lima: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing. **Ezequiel López-Rubio:** Conceptualization, Funding acquisition, Project administration, Supervision, Validation, Writing – review & editing, Formal analysis. **Juan Miguel Ortiz-de-Lazcano-Lobato:** Conceptualization, Supervision, Validation, Writing – review & editing, Formal analysis. **José David Fernández-Rodríguez:** Conceptualization, Investigation, Writing – review & editing, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is partially supported by the Ministry of Science and Innovation of Spain [grant number PID2022-136764OA-I00], project name Automated Detection of Non Lesional Focal Epilepsy by Probabilistic Diffusion Deep Neural Models. It is also partially supported by the Autonomous Government of Andalusia (Spain) under project UMA18-FEDERJA-084, project name Detection of anomalous behaviour agents by deep learning in low-cost video surveillance intelligent systems. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise, and assistance provided by the SCBI (Supercomputing and Bioinformatics) centre of the University of Málaga. The authors acknowledge the funding from the Universidad de Málaga, Spain. No conflict of interest has been declared by the authors.

Data availability

Data will be made available on request.

References

[1] A. Foucart, O. Debeir, C. Decaestecker, Shortcomings and areas for improvement in digital pathology image segmentation challenges, Comput. Med. Imaging Graph. (2022) 102155.

[2] Y. Yu, C. Wang, Q. Fu, R. Kou, F. Huang, B. Yang, T. Yang, M. Gao, Techniques and challenges of image segmentation: A review, Electron. 12 (5) (2023) http://dx.doi.org/10.3390/electronics12051199.

[3] C.M. de Melo, A. Torralba, L. Guibas, J. DiCarlo, R. Chellappa, J. Hodgins, Next-generation deep learning based on simulators and synthetic data, Trends Cogn. Sci. (2021).

[4] Y. Chen, W. Li, X. Chen, L.V. Gool, Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1841–1850.

[5] Y. Chen, Z. Yan, Y. Zhu, A comprehensive survey for generative data augmentation, Neurocomputing 600 (2024) 128167, http://dx.doi.org/10.1016/j.neucom.2024.128167.

[6] F.-A. Croitoru, V. Hondru, R.T. Ionescu, M. Shah, Diffusion models in vision: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 45 (9) (2023) 10850–10869, http://dx.doi.org/10.1109/TPAMI.2023.3261988.

[7] Y.-H. Park, M. Kwon, J. Choi, J. Jo, Y. Uh, Understanding the latent space of diffusion models through the lens of riemannian geometry, Adv. Neural Inf. Process. Syst. 36 (2023) 24129–24142.

[8] M. Chen, S. Mei, J. Fan, M. Wang, Opportunities and challenges of diffusion models for generative AI, Natl. Sci. Rev. 11 (12) (2024) nwae348.

[9] B. Liu, S. Shao, B. Li, L. Bai, Z. Xu, H. Xiong, J. Kwok, S. Helal, Z. Xie, Alignment of diffusion models: Fundamentals, challenges, and future, 2024, arXiv preprint arXiv:2409.07253.

[10] M. Sannidhan, G.A. Prabhu, D.E. Robbins, C. Shasky, Evaluating the performance of face sketch generation using generative adversarial networks, Pattern Recognit. Lett. 128 (2019) 452–458.

[11] K.E. Ak, J.H. Lim, J.Y. Tham, A.A. Kassim, Semantically consistent text to fashion image synthesis with an enhanced attentional generative adversarial network, Pattern Recognit. Lett. 135 (2020) 22–29.

[12] D. Li, X. Nie, X. Li, Y. Zhang, Y. Yin, Context-related video anomaly detection via generative adversarial network, Pattern Recognit. Lett. 156 (2022) 183–189.

[13] O. Kafri, O. Patashnik, Y. Alaluf, D. Cohen-Or, Stylefusion: Disentangling spatial segments in stylegan-generated images, ACM Trans. Graph. 41 (5) (2022) 1–15.

[14] X. He, B. Wandt, H. Rhodin, Ganseg: Learning to segment by unsupervised hierarchical image generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1225–1235.

[15] D. Pakhomov, S. Hira, N. Wagle, K.E. Green, N. Navab, Segmentation in style: Unsupervised semantic image segmentation with stylegan and CLIP, 2021, arXiv preprint arXiv:2107.12518.

[16] Y. Zhang, H. Ling, J. Gao, K. Yin, J.-F. Lafleche, A. Barriuso, A. Torralba, S. Fidler, Datasetgan: Efficient labeled data factory with minimal human effort, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10145–10155.

[17] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2021) 4217–4228.

[18] D. Li, H. Ling, S.W. Kim, K. Kreis, S. Fidler, A. Torralba, BigDatasetGAN: Synthesizing ImageNet with pixel-wise annotations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 21330–21340.

[19] S. Rizvi, P. Cicalese, S. Seshan, S. Sciascia, J. Becke, H. Nguyen, Histopathology datasetgan: Synthesizing large-resolution histopathology datasets, in: 2022 IEEE Signal Processing in Medicine and Biology Symposium, SPMB, IEEE, 2022, pp. 1–4.

[20] B. Lutnick, N. Lucarelli, P. Sarder, Generative modeling of histology tissue reduces human annotation effort for segmentation model development, in: Medical Imaging 2023: Digital and Computational Pathology, Vol. 12471, SPIE, 2023, pp. 453–457.

[21] H. Chu, E. Agapaki, L. Deng, Tunnel-crack-datasetgan: A multi-scene deep domain adaptive crack generator for tunnel-lining crack segmentation, in: Expanding Underground-Knowledge and Passion To Make a Positive Impact on the World, CRC Press, 2023, pp. 2618–2626.

[22] B. Song, J. Wang, X. Wang, T. Zeng, D. Li, Pixel-wise annotated and high-quality synthesized image datasets for semi-supervised truck segmentation with limited raw images, Autom. Constr. 158 (2024) 105197, http://dx.doi.org/10.1016/j.autcon.2023.105197.

[23] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, T. Aila, Training generative adversarial networks with limited data, Adv. Neural Inf. Process. Syst. 33 (2020) 12104–12114.

[24] Z. Fan, V. Kelkar, M.A. Anastasio, H. Li, Application of datasetgan in medical imaging: preliminary studies, in: Medical Imaging 2022: Image Processing, 12032, SPIE, 2022, pp. 452–458.

[25] D. Baranchuk, I. Rubachev, A. Voynov, V. Khrulkov, A. Babenko, Label-efficient semantic segmentation with diffusion models, 2021, arXiv preprint arXiv:2112.03126.

[26] W. Wu, Y. Zhao, H. Chen, Y. Gu, R. Zhao, Y. He, H. Zhou, M.Z. Shou, C. Shen, Datasetdm: Synthesizing data with perception annotations using diffusion models, Adv. Neural Inf. Process. Syst. 36 (2023) 54683–54695.

[27] A. Xu, M.I. Vasileva, A. Dave, A. Seshadri, Handsoff: Labeled dataset generation with no additional human annotations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7991–8000.

[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[29] G. Chen, P. Chen, Y. Shi, C.-Y. Hsieh, B. Liao, S. Zhang, Rethinking the usage of batch normalization and dropout in the training of deep neural networks, 2019, arXiv preprint arXiv:1905.05928.

[30] L. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell. 12 (10) (1990) 993–1001, http://dx.doi.org/10.1109/34.58871.