A Comparative Study of an AI Model's Robustness to Synthetic Data in Solving the Problem of Color Image Classification

Marina Barulina *, Sergey Okunkov, Ivan Ulitin

Posted Date: 7 April 2026

doi: 10.20944/preprints202604.0294.v1

Keywords: deep learning; unbalanced dataset; augmentation; synthetic data; multiclass classification; metrics boosting method; visual transformer; robustness; color image classification

 doi:10.20944/preprints202604.0294.v1

Article

A Comparative Study of an AI Model's Robustness to Synthetic Data in Solving the Problem of Color Image Classification

Marina Barulina [1,2,*], Sergey Okunkov [2,3] and Ivan Ulitin [2,3]

[1] Saint Petersburg State University, Saint-Petersburg, Russia

[2] Institute of Physical and Mathematical Sciences, Perm State National Research University, Bukireva Street, 15, 614990 Perm, Russia

[3] Institute of Precision Mechanics and Control of the Russian Academy of Sciences, 24, ul. Rabochaya, 410028 Saratov

* Correspondence: m.a.barulina@spbu.ru

Abstract

This study examines the impact of data augmentation on machine learning perfor-mance, focusing on how synthetic data influences various neural network architec-tures. Common issues such as limited data, class imbalance, and poor coverage often lead to low model metrics, and data augmentation is frequently used to address these problems. The research aims to identify the optimal proportion of synthetic data, assess its effects across different architectures, and analyze the impact of augmenting only specific classes in a multi-class medical image classification task. Twelve widely used architectures were selected for the experiments, including classical convolutional networks, visual transformers, and the hybrid ConvNeXt model. Results showed that no universal optimal augmentation ratio exists, as model robust-ness to synthetic data varies, even within the same architecture family. Transformer and hybrid models demonstrated greater stability, while convolutional networks exhibited inconsistent behavior, likely due to higher sensitivity to data bias.

Keywords: deep learning; unbalanced dataset; augmentation; synthetic data; multiclass classification; metrics boosting method; visual transformer; robustness; color image classification

1. Introduction

High-quality data is the foundation of any effective artificial intelligence system. The performance of such systems depends heavily on the characteristics of the dataset used, including sample size, data balance, example diversity, number of classes, presence of bias, and the overall representativeness of the data within the studied domain. Consequently, poor model performance in machine learning and deep learning tasks often stem from insufficient or low-quality source data.

To address these limitations, data augmentation techniques are frequently applied to generate synthetic data, offering a cost-effective way to expand datasets and im-prove model generalization. This approach assumes that neural networks, even small changes in input data — down to a single altered bit — can provide new learning opportunities.

Machine learning researchers have always been interested in this topic. However, most studies in this area analyze the impact of synthetic data on specific architectures and tasks, focusing on the augmentation algorithms themselves and failing to address key experimental design questions. The main idea behind these studies is that moderate use of this technique yields a modest increase in prediction quality.

So, article [1] shows that by using simple image conversion algorithms in different tasks and CV models, a 1-3% gain can be achieved, which can be a very good improvement relative to the effort

 doi:10.20944/preprints202604.0294.v1

expended. However, it is worth noting that this article also provides examples of a slight degradation in model quality of 0.1-0.3%, so such methods should be used with caution.

By analogy, the authors of article [2] come to similar conclusions. However, in their examples, no degradation in quality was observed, and augmentation had only a positive effect on the models. This is particularly noticeable for segmentation tasks. At the same time, unlike in their previous work, the authors consider in more detail the influence of each class of methods used on the result.

Article [3] uses more complex approaches to data augmentation, which allows the authors to achieve a 3-3.5% increase in metrics for the classification task, which is significantly higher than in the works described above. This means that by improving the methodology for obtaining synthetic data, it is possible to achieve much higher results.

Article [4] discusses an approach to increasing the sample size using synthetic data in a highly specialized field (medicine), where some image augmentation methods are completely inapplicable.

In work [5], researchers aim to study the impact of synthetic data generated using a GAN model used to train ResNet-18 on classification results. As a result, there is a significant increase in the quality of the final prediction for some complex samples. However, the authors use simple datasets that have long been out of use as initial samples, which indicates the complexity of applying this approach to real data.

The paper [6] describes the complexity of applying synthetic data in the domain of NLP, since there are no algorithms for this domain that simulate real problems in data that may arise due to external factors. It is also worth noting that today all these approaches are outdated and have given way to conventional text generation using LLM.

The authors of article [7] focus on the results of applying augmentation when training classical machine learning models (SVM, KNN, RF, DT). They show that synthetic data can have a very positive effect when training models that are not related to neural networks. Target metrics increased by more than 3.5%, which is a record among all the articles reviewed.

However, despite its popularity and demonstrated benefits, data augmentation is not a universal solution. Suitable algorithms are not available for all data types or tasks, and in some cases, the generation of synthetic examples can distort key features or alter target labels, leading to degraded model performance. Furthermore, the effects of augmentation on different models remain insufficiently understood.

Several practical questions arise when applying augmentation techniques:

1. What is the optimal percentage of synthetic data beyond which performance gains cease or begin to decline?
2. How does synthetic data affect different neural network architectures?
3. How does targeted augmentation of minority classes affect model behavior?
4. How sensitive are models to the choice of augmentation strategy?
5. In which task scenarios is this approach trustworthy?

Our study looks at how several popular neural network architectures react to synthetic data concentrated in rare classes, using a real-world color image classification task as the main experimental setting. Specifically, the research addresses the first three questions outlined above. We will consider a model robust to synthetic data if its performance metrics vary little and remain consistently high across a wide range of synthetic data shares, demonstrating low variance between runs and augmentation levels. Note that in this work, synthetic data refers to classically augmented images obtained via geometric and photometric transformations, not to samples generated by GANs or diffusion models.

 doi:10.20944/preprints202604.0294.v1

2. Materials and Methods

2.1 Datasets

Two samples were used for the experiments: the main sample for conducting the experiments and the validation sample for confirming that the results and conclusions obtained were not accidental.

Both samples were transformed using the following methodology (Figure 1):

1. Initial data: The initial samples were highly unbalanced sets of images divided into several classes.
2. Data balancing: Each sample was reduced to the number of examples in the smallest class, which ensured data balance.
3. Sample division: The balanced set was divided into training and test samples in a 4:1 ratio. The training sample was used to train the models, and the test sample was used to evaluate their quality.
4. Creating samples for training: The training sample was supplemented with synthetic data at 10% increments relative to the size of the original sample. To do this, pre-generated data (based on the methods described in section 1.6) was added to the small classes, and examples from the original sample were added to the rest. At each stage of supplementation, the results were recorded in a separate sample, named according to the proportion of added data (0%, 10%, 20%, etc.).

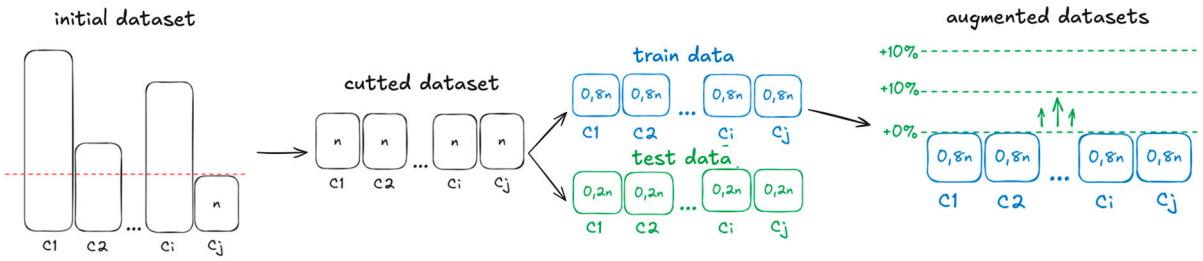

Figure 1. *Creating experiment datasets.*

The main sample consisted of a set of 20,971 RGB images obtained by crawling Instagram accounts and labeled by a specialist into seven classes in accordance with known standards [8]. The distribution of classes in this sample was highly unbalanced: C0 (11.89%), C1 (26.2%), C2 (13.64%), C3 (32.66%), C4 (11.38%), C5 (2.18%), and C6 (2.04%). Examples of each class of samples are presented in Figure 2.

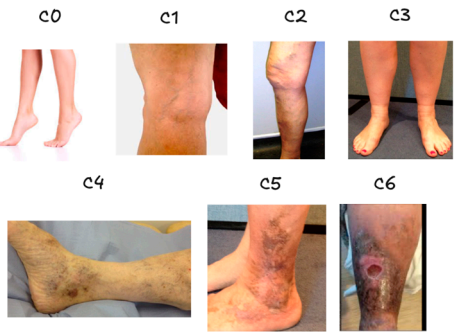

Figure 2. *Examples of each class of samples in main dataset.*

 doi:10.20944/preprints202604.0294.v1

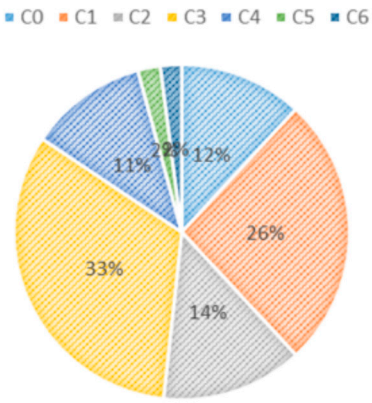

Figure 3. *Class distribution in main dataset.*

Table 1. Class distribution of images of the main dataset.

Class	C0	C1	C2	C3	C4	C5	C6
Number of images	2494	5495	2861	6850	2386	458	427

An open dataset of skin lesions from Kaggle [9] was used as a validation sample. It contained 13,202 RGB images of 14 different skin diseases: actinic keratosis, basal cell carcinoma, benign keratosis-like lesions, basal cell carcinoma, benign keratosis-like lesions, chickenpox, cowpox, dermatofibroma, HFMD, healthy skin, measles, melanocytic nevi, melanoma, monkeypox, squamous cell carcinoma, and vascular lesions. However, this dataset contained many synthetic examples. After preprocessing and removing obviously generated examples (like augmented classes with few real examples), the number of images was 6787, the number of classes was 8, and their distribution is shown in Figure 18, where C0 is actinic keratosis, C1 is basal cell carcinoma, C2 is benign keratosis-like lesions, C3 is dermatofibroma, C4 is melanocytic nevi, C5 is melanoma, C6 is squamous cell carcinoma, and C7 is vascular lesions. Examples of each class of samples are presented in Figure 4.

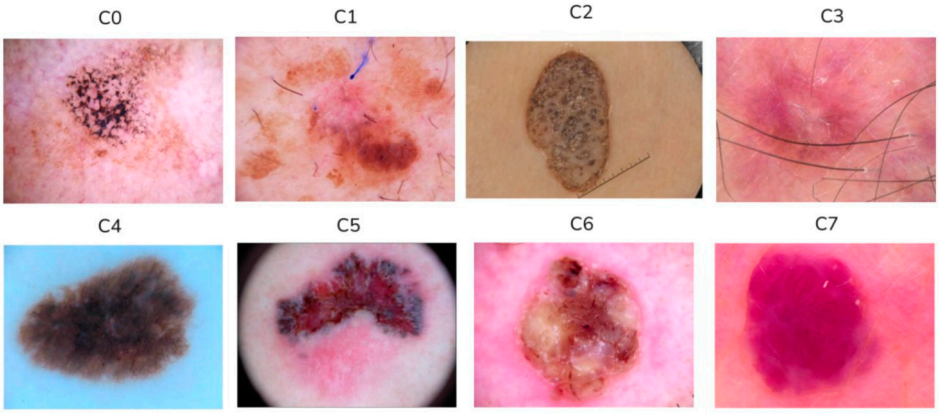

Figure 4. *Examples of each class of samples in validation dataset.*

 doi:10.20944/preprints202604.0294.v1

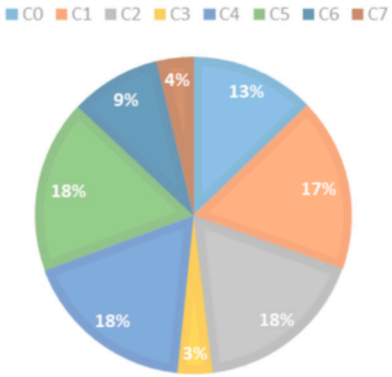

Figure 5. *Class distribution in validation dataset.*

Table 2. Class distribution of images of the validation dataset.

Class	C0	C1	C2	C3	C4	C5	C6	C7
Number of images	882	1154	1221	204	1222	1221	611	271

2.2. Deep Learning Neural Networks

We trained the following 12 pretrained different contemporary architectures of deep learning neural networks on the obtained datasets: ResNet34, ResNet50, ResNet101 [10], VIT-base-patch16-224, DeIT-base-patch16-224, BeIT-base-patch-224 [11], DenseNet161, DenseNet201 [12], VGG19 [13], Inception [14], Xception [15], ConvNeXT [16]. To ensure fairness of the experiment and exclude randomization, the training process was run 20 times for each model. Each training run involved shuffling the data when splitting it into training and validation dataset. Our goal was to study how synthetic data in rare classes would affect each of the architectures, whether there would be any differences, or whether the results would be roughly similar.

The VGG architecture (Figure 6) is one of the first widely adopted deep convolutional neural networks, introduced to systematically study the effect of depth on recognition performance. Its main design principle is to build a hierarchical representation of images by stacking uniform convolutional blocks, each composed of multiple 3×3 convolutional layers followed by 2×2 max-pooling, and then to classify these features using a series of fully connected layers. This modular, depth-based structure enables the network to learn increasingly complex visual patterns in successive blocks, making VGG a popular choice for transfer learning and feature-extraction in downstream computer-vision tasks.

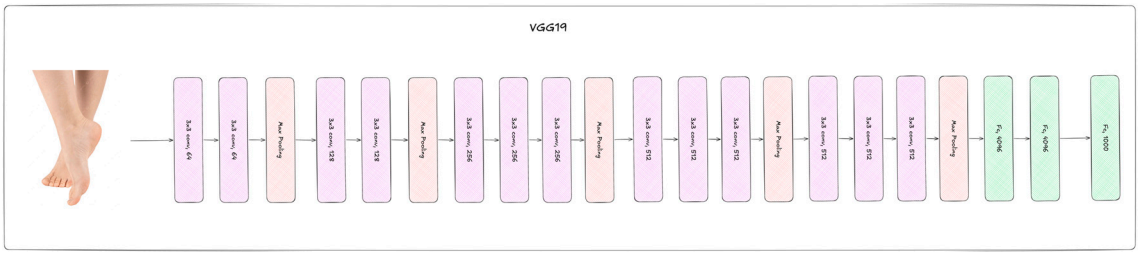

Figure 6. *Architecture of VGG19.*

ResNet (Residual Neural Network) models are the most popular baseline in computer vision. ResNet is a convolution neural network in which the weight layers learn residual functions with reference to the layer inputs. The fundamental difference between ResNet and classical convolutional deep learning models is shortcut connections. Thanks to them, ResNet can contain a lot of layers while bypassing the vanishing gradient problem, which helps to extract richer features from input images. The architecture of the residual block is shown in Fig. 7.

 doi:10.20944/preprints202604.0294.v1

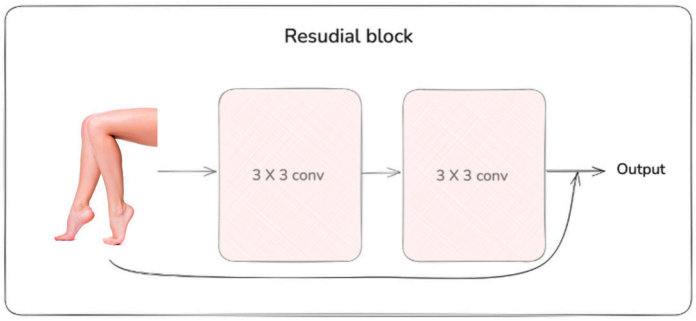

Figure 7. *Architecture of residual block.*

The Inception architecture debuted in Google's GoogLeNet (Inception v1) by Szegedy et al. (2015), a 22-layer CNN that hit a top-5 error of 6.67% on ImageNet ILSVRC 2014 using just 5 million parameters—about 7x fewer than VGGNet.Its key innovation, inception modules, process input feature maps via parallel branches: a 1×1 convolution for reduction, 1×1, 3×3, and 5×5 convolutions for multi-scale features, plus a 3×3 max-pooling branch followed by 1×1 projection. Branch outputs concatenate channel-wise, boosting representational power while keeping spatial dimensions intact and parameters low through 1×1 bottlenecks that cut costs by 70–90%. The architecture of the inception block is shown in Fig. 8.

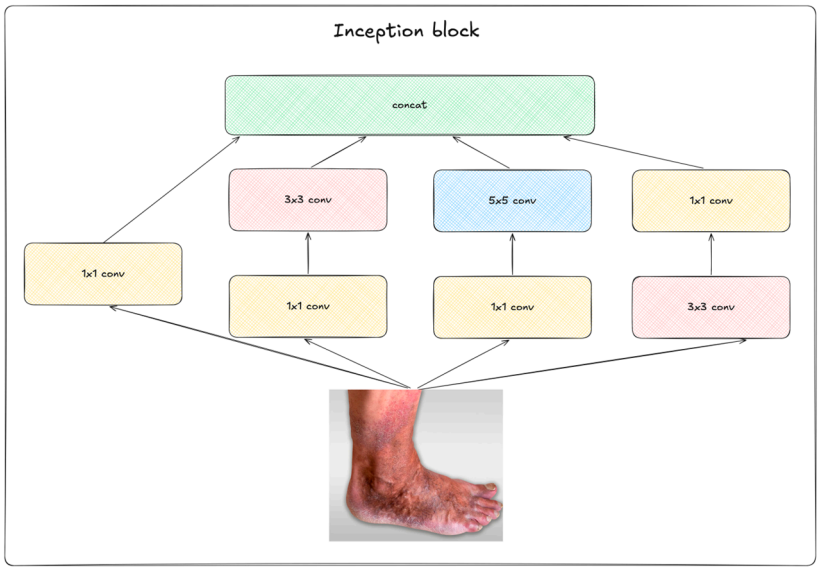

Figure 8. *Architecture of inception block.*

Xception, proposed by Chollet (2017), evolves Inception by fully separating spatial and channel-wise convolutions via depth wise separable convolutions in xception blocks, based on the extreme inception hypothesis that channel mixing works better independently from spatial filtering. Each block sequences three operations: a 1×1 pointwise convolution for channel expansion/reduction, depth wise 3×3 convolution (one filter per input channel for spatial features), and another 1×1 pointwise convolution to recombine channels. This linear stack replaces Inception's parallel branches, repeated 8 times per module across entry, middle, and exit flows in the full network. Efficiency stems from depthwise separable ops needing 8–9x fewer parameters than standard convolutions for the same mapping. The architecture of the inception block is shown in Fig. 9.

 doi:10.20944/preprints202604.0294.v1

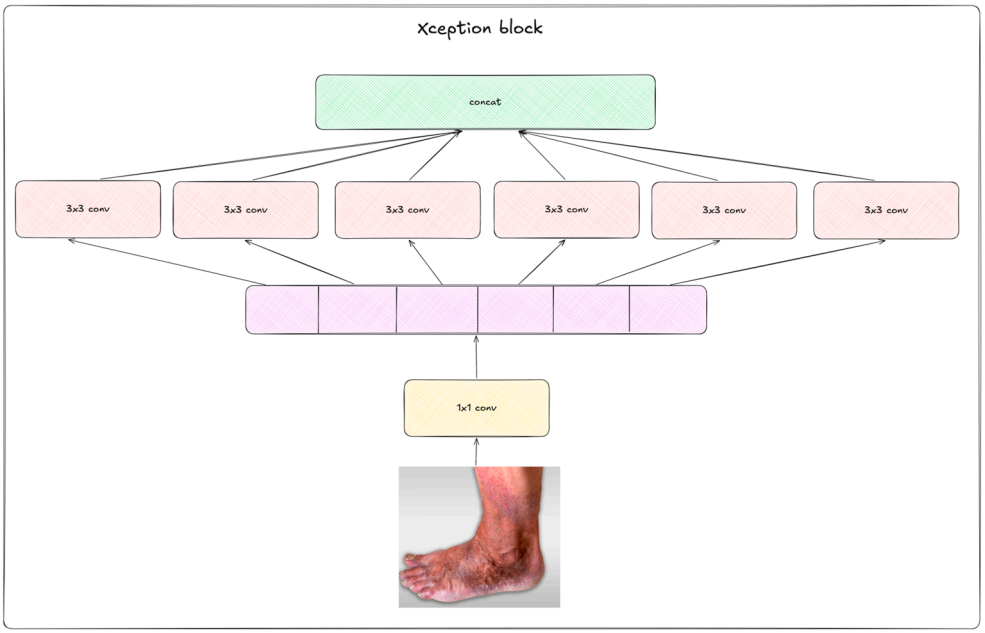

Figure 9. *Architecture of xception block.*

DenseNet is modified ResNet. DenseNet establishes a dense connection between all the front and back layers. Another important feature of DenseNet is the implementation of feature reuse via feature bonding on a channel. Thanks to these features, DenseNet can learn well from small datasets. Example architecture of dense block is shown in Figure 10.

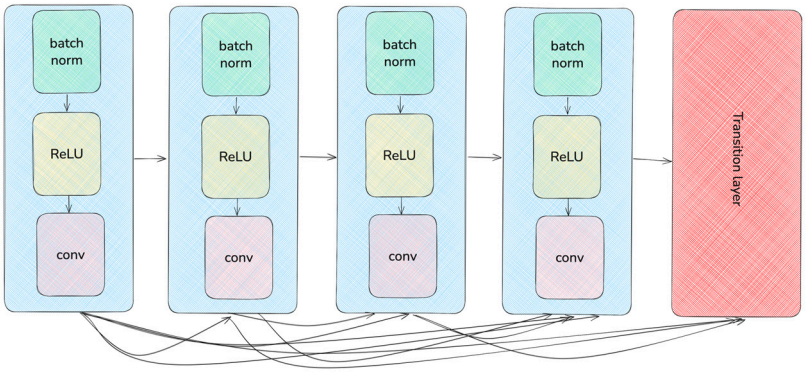

Figure 10. *Architecture of dense block.*

Visual image transformer (VIT), Data efficient image transformer (DeIT) and Bidirectional encoder image transformer (BEIT) are state of the art (SOTA) architectures in computer vision at this moment. They are based on transformer architecture, which was created for natural language processing tasks. The main idea of these architectures is to use not pixels as layers, but fixed-size image pieces called tokens or patches. These tokens processing like texts tokens on NLP, using attention layers for getting image embeddings. Example architecture of image transformer neural networks is shown in Figure 11.

 doi:10.20944/preprints202604.0294.v1

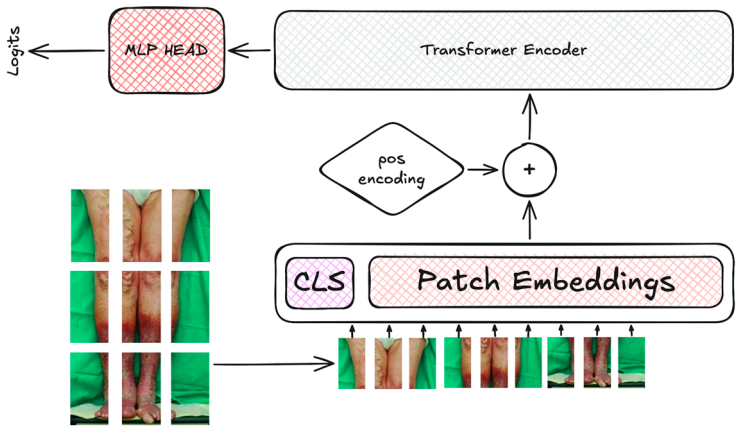

Figure 11. *Example of image transformer architecture.*

ConvNeXt, introduced by Liu et al. (2022), modernizes CNNs to rival vision transformers like Swin-Transformer on image classification and vision tasks, using pure convolutions with transformer-inspired designs—no self-attention needed. It adopts a staged macro-architecture with four downsampling stages, each stacking ConvNeXt blocks in a transformer-like compute distribution to build global representations as resolution drops and channels grow. The core ConvNeXt block applies channel-wise LayerNorm, a large-kernel 7×7 depthwise convolution for spatial context, and a pointwise MLP (two 1×1 convolutions with GELU) for cross-channel mixing, plus residuals with stochastic depth. A patchify stem starts with a 4×4 stride-4 convolution and LayerNorm, mimicking transformer patch embeddings to coarsen input from 224×224×3. Stages alternate blocks and downsampling, ending in global average pooling and a linear classifier; it matches Swin performance at similar FLOPs with better throughput for detection and segmentation. Architecture of ConvNeXT block is shown in Figure 12.

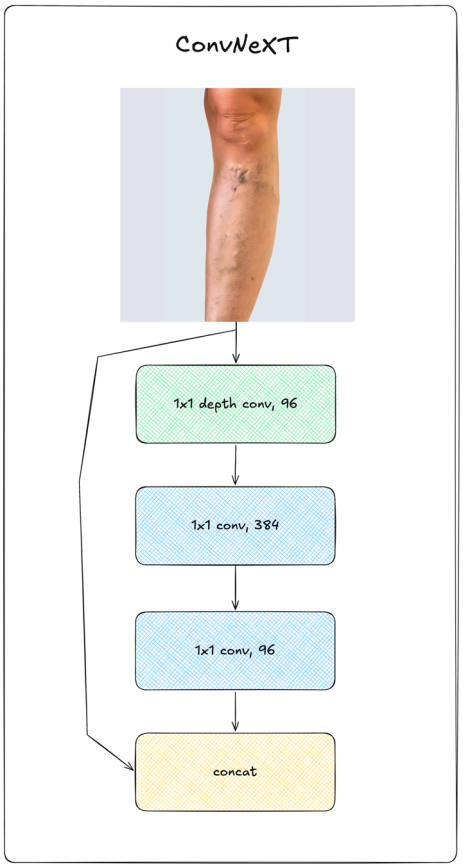

Figure 12. *Architecture of ConvNeXT block.*

 doi:10.20944/preprints202604.0294.v1

2.3. Augmentation Methods

The following transformations were used:

1. Shift, Scale, and Rotate (Affine Transformations). This transformation randomly applies to a combination of affine operations, including translation, scaling, and rotation. The scaling factor range was set to (-0.5, 0.5), meaning that each image could be randomly scaled up or down by up to 50% relative to its original size. The shift range for both horizontal and vertical translation was (-0.05, 0.05), corresponding to a potential displacement of up to 5% of the image size along each axis. The rotation range was defined as (-30°, 30°), allowing images to be randomly rotated within this angular range. The probability of applying this transformation was 0.6. This method helps simulate variations in object position, orientation, and scale that might naturally occur in real-world conditions.

2. Random Crop. This operation extracts a random portion of each image while maintaining a fixed output size of 224 × 224 pixels. It helps the model become invariant to the spatial positioning of key features, ensuring more robust feature learning and reducing overfitting image boundaries.

3. Horizontal Flip. Images were horizontally flipped to generate their mirrored counterparts, thereby increasing variation in object orientation. The probability of applying this transformation was 0.5. Such flipping is especially useful when the target objects have no inherent directional bias.

4. RGB Shift. To simulate variations in lighting and sensor characteristics, random shifts were applied independently to each color channel (Red, Green, and Blue). The range of value change for each channel was (-15, 15). The probability of applying the RGB Shift was 0.6. This technique improves robustness to color and illumination changes in different environments.

5. Random Brightness and Contrast Adjustment. This transformation randomly varies from the brightness and contrast of an image to model lighting conditions under which the data might be captured. The probability of application was set to 0.7. This adjustment helps prevent the model from overfitting specific lighting conditions in the training set.

2.4. Training Procedure

Dataset preprocessing included only resizing input images to 224x224 px. For VIT-base-patch16-384 the input images were resized to 384x384 px. The constructed datasets were divided into training (80%) and test (20%) datasets.

The training environment was CPU AMD Ryzen 5 5600 GPU NVIDIA RTX 2070 Super RAM 16 GB.

The training parameters for all models are shown in Table 4. The rest of the parameters had default values. It is worth noting that such a spread of the batch size parameter is associated with the size of the neural network models, as well as the optimization of their training in a specific library.

Table 4. The training parameters for all considered models.

	Num of params	Image size	Batchsize	Num of hidden layers	Optimizer	Learning rate
ResNet34	21.7M	224	64	34	Adam	$1 \cdot 10^{-4}$
ResNet50	25.5M	224	64	50	Adam	$1 \cdot 10^{-4}$
ResNet101	44.5M	224	64	101	Adam	$1 \cdot 10^{-4}$
VGG19	143.6M	224	16	19	Adam	$1 \cdot 10^{-4}$
DenseNet161	28.7M	224	8	161	Adam	$1 \cdot 10^{-4}$
DenseNet201	20M	224	8	201	Adam	$1 \cdot 10^{-4}$
Inception_v3	27.2M	224	16	48	Adam	$1 \cdot 10^{-4}$
Xception	23M	224	8	71	Adam	$1 \cdot 10^{-4}$
VIT	86.4M	224	32	12	AdamW	$5 \cdot 10^{-5}$
DeIT	86.4M	224	32	12	AdamW	$5 \cdot 10^{-5}$

 doi:10.20944/preprints202604.0294.v1

BeIT	86.9M	384	2	12	AdamW	$5 \cdot 10^{-5}$
ConvNeXT	28M	224	8	36	AdamW	$5 \cdot 10^{-5}$

2.5. Metrics

To evaluate the quality of model training we used the following metrics - Accuracy, Precision, Recall, F1.

Accuracy measures the number of correct predictions made by a model in relation to the total number of predictions made:

$$Accuracy = \frac{Correct\ classification}{All\ classification} \quad (1)$$

Note that the accuracy could be misleading for imbalanced datasets. But since we formed balanced datasets, we can use this metric for the quality evaluation.

The precision measures the percentage of predictions made by the model that are correct:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

The recall (true positive rate, TPR, sensitivity) measures the percentage of relevant data points that were correctly identified by the model:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

The F1 score is a machine learning evaluation metric that measures a model's accuracy. The F1 score is defined as the harmonic mean of precision and recall. This metric computes how many times a model made a correct prediction across the entire dataset:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (4)$$

3. Results

The metrics obtained for the considered neural networks on the generated datasets are shown in Table 5, Table 6 for common metrics and on Figure A1-A24 for class metrics.

Table 5. Accuracy metrics for main dataset.

%, synthetic data	VGG 19	ResNet 34	ResNet 50	ResNet101	Xception	Inception	DenseNet 121	DenseNet 201	VIT	DeIT	BeIT	ConvNeXT
0	0,47	0,57	0,58	0,57	0,6	0,6	0,62	0,63	0,62	0,62	0,64	0,62
10	0,5	0,58	0,58	0,58	0,6	0,61	0,62	0,63	0,66	0,63	0,65	0,63
20	0,56	0,58	0,6	0,59	0,61	0,61	0,64	0,64	0,67	0,65	0,66	0,63
30	0,56	0,58	0,61	0,61	0,62	0,62	0,64	0,64	0,67	0,65	0,67	0,64
40	0,56	0,58	0,58	0,62	0,63	0,61	0,63	0,65	0,67	0,65	0,67	0,64
50	0,55	0,59	0,59	0,6	0,64	0,58	0,64	0,64	0,68	0,65	0,68	0,64
60	0,56	0,6	0,6	0,61	0,6	0,6	0,64	0,64	0,68	0,66	0,68	0,64
70	0,6	0,6	0,64	0,63	0,6	0,62	0,65	0,65	0,68	0,65	0,68	0,65
80	0,58	0,61	0,63	0,6	0,63	0,6	0,65	0,65	0,7	0,65	0,7	0,66
90	0,55	0,58	0,61	0,63	0,63	0,63	0,64	0,65	0,69	0,65	0,68	0,64
100	0,56	0,58	0,58	0,62	0,62	0,6	0,65	0,65	0,69	0,65	0,68	0,64

 doi:10.20944/preprints202604.0294.v1

Table 6. Accuracy metrics for validation dataset.

%, synthetic data	VGG 19	ResNet 34	ResNet 50	ResNet101	Xception	Inception	DenseNet121	DenseNet201	VIT	DeIT	BeIT	ConvNeXT
0	0,71	0,76	0,77	0,78	0,8	0,79	0,8	0,8	0,82	0,79	0,82	0,82
10	0,72	0,78	0,78	0,78	0,8	0,79	0,81	0,81	0,83	0,8	0,82	0,82
20	0,73	0,79	0,79	0,79	0,79	0,78	0,8	0,8	0,84	0,81	0,84	0,83
30	0,73	0,79	0,79	0,79	0,78	0,78	0,8	0,8	0,84	0,81	0,83	0,82
40	0,73	0,79	0,79	0,8	0,79	0,77	0,79	0,78	0,84	0,81	0,83	0,82
50	0,71	0,78	0,77	0,78	0,77	0,78	0,79	0,79	0,84	0,81	0,83	0,82
60	0,7	0,78	0,78	0,78	0,78	0,79	0,78	0,79	0,84	0,81	0,82	0,81
70	0,7	0,78	0,79	0,8	0,77	0,78	0,8	0,81	0,84	0,81	0,82	0,81
80	0,7	0,77	0,77	0,78	0,77	0,78	0,79	0,78	0,84	0,81	0,82	0,81
90	0,69	0,76	0,77	0,78	0,76	0,78	0,76	0,78	0,84	0,82	0,82	0,8
100	0,72	0,77	0,78	0,79	0,77	0,77	0,78	0,77	0,84	0,81	0,81	0,8

Tables 5 and 6 illustrate the differences in how various architectures respond to synthetic data augmentation, particularly in terms of overall accuracy. Transformer-based models and the ConvNeXT hybrid architecture exhibit pronounced parabolic dynamics: performance metrics rise to an optimal enrichment threshold, after which they plateau or begin to decline, consistent with the diminishing returns effect from excessive synthetic perturbations. In contrast, the behavior of traditional CNNs is much less predictable. Their metrics change in a non-monotonic way and jump up and down as the share of synthetic data grows, without a clear overall trend. This may reflect the fact that CNNs are more sensitive to changes in the data distribution and rely heavily on local feature patterns encoded in their inductive bias. As can be seen from Figures A1–A24, the same effect appears at the level of per-class metrics. For many CNNs the curves are noisy and unstable, so it is hard to speak about a single, consistent pattern. By comparison, the curves for vision transformers and ConvNeXt are noticeably smoother and show smaller variance across augmentation levels, which suggests that these architectures are less affected by artifacts introduced by synthetic data.

4. Discussion

Experiments showed that different architectures respond to synthetic images very differently, meaning there is no universal augmentation strategy.

The ViT model demonstrated a steady increase in quality as the proportion of synthetic data increased, but only up to a certain point—around 80%. Beyond this point, further addition of synthetic data was ineffective. The model clearly became saturated.

Overall, models from the Vision Transformer family were the least sensitive to augmentation—their quality remained relatively stable regardless of its volume.

Convolutional networks (CNNs), on the other hand, exhibited significantly more instability. This instability was particularly evident at the individual class level, as per-class metrics fluctuated more than aggregated metrics. The more complex the classification task, the more pronounced this effect became, as the variance of the metrics increased with the complexity of the task. On the validation dataset, all models performed generally better than when trained solely on the original data, but at the expense of greater variability across classes and runs.

The ConvNeXt architecture deserves special attention. In terms of robustness to synthetic data, it proved comparable to transformer models, behaving more like a Vision Transformer than a classic convolutional network. This is consistent with its nature, as ConvNeXt was originally designed with the idea of transformers in mind, which likely determines its stability during augmentation.Furthermore, aggregate performance metrics (e.g., overall accuracy or macro-average

 doi:10.20944/preprints202604.0294.v1

scores) may not fully reflect the impact of synthetic data. Therefore, conclusions about the comparative performance of augmentation strategies based solely on such metrics should be treated with caution.

We note that this study has several limitations that should be considered when interpreting the results. First, both datasets were artificially balanced by reducing the sample size of all classes to the size of the rarest class. While this choice simplifies the interpretation of metrics and prevents the dominance of majority classes, it also reduces the effective sample size and may limit the absolute performance achievable by all models. Second, in the validation dataset, we manually removed "obviously synthetic" samples, which introduces a certain degree of subjectivity and may not perfectly reflect the distribution of real-world data. Third, the concept of "synthetic data" in this work is limited to classic geometric and photometric augmentations; GAN-based image generation or diffusion was not used, so the results cannot be directly generalized to all types of synthetic data. Finally, although we reduced the influence of randomness by running each configuration 20 times, all experiments were conducted within a single training configuration (hardware, library implementation, hyperparameters), and we considered only two specific image classification tasks. Therefore, the robustness patterns observed here for CNNs, Vision Transformers, and ConvNeXt may differ for other domains, datasets, or training modes.

5. Conclusions

The experiments conducted have shown that, for most architectures, using a relatively small amount of synthetic data during training can be beneficial. In particular, when the training and validation sets were expanded by 10% with augmented examples, all the models considered demonstrated a consistent (though often moderate) improvement across the evaluated metrics. This suggests that a 10% augmentation level can be regarded as a practical and safe volume of data augmentation for classification tasks rather than a universally optimal proportion. At the same time, we observe that different model families respond differently to further increases in the proportion of synthetic data: vision transformers and ConvNeXt models generally maintain fairly stable performance across a wide range of augmentation levels, whereas convolutional networks exhibit more pronounced fluctuations in metrics as the share of synthetic data grows. Further research should explore the impact of synthetic data obtained using more sophisticated approaches, such as GAN or diffusion models.

Author Contributions: Conceptualization, M.B. and S.O.; methodology, M.B.; software, S.O.; validation, S.O. and I.U.; formal analysis, I.U.; investigation, S.O.; resources, M.B.; data curation, M.B.; writing—original draft preparation, S.O. and I.U.; writing—review and editing, M.B.; visualization, S.O.; supervision, M.B.; project administration, M.B.; funding acquisition, M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Economic Development of the Russian Federation (Agreement No. 139-15-2025-007, dated April 16, 2025; ID: 000000C313925P3O0002).

Data Availability Statement: Kaggle Skin Dataset is available online: https://www.kaggle.com/datasets/ahmedxc4/skin-ds (accessed on 29 March 2026).

Acknowledgments: Not applicable

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

 doi:10.20944/preprints202604.0294.v1

Appendix A

Figure A1. *Metrics of VGG19 on main dataset.*

Figure A2. *Metrics of VGG19 on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A3. *Metrics of ResNet34 on main dataset.*

Figure A4. *Metrics of ResNet34 on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A5. *Metrics of ResNet50 on main dataset.*

Figure A6. *Metrics of ResNet50 on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

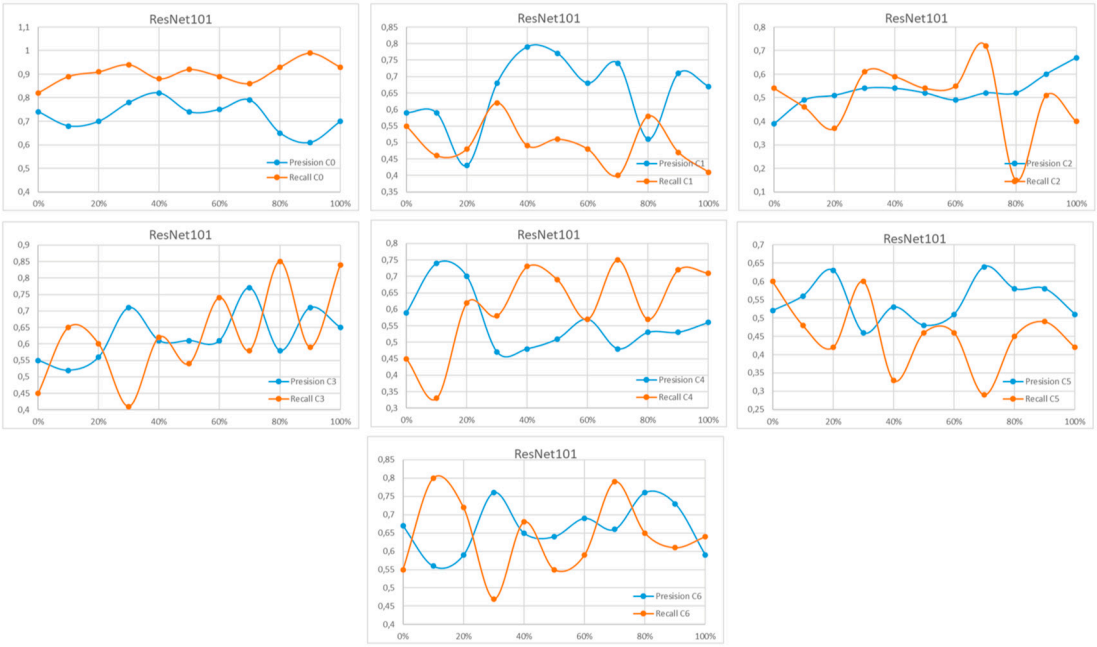

Figure A7. *Metrics of ResNet101 on main dataset.*

Figure A8. *Metrics of ResNet101 on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A9. *Metrics of Inception on main dataset.*

Figure A10. *Metrics of Inception on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A11. *Metrics of Xception on main dataset.*

Figure A12. *Metrics of Xception on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A13. *Metrics of DenseNet121 on main dataset.*

Figure A14. *Metrics of DenseNet121 on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A15. *Metrics of DenseNet201 on main dataset.*

Figure A16. *Metrics of DenseNet121 on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A17. *Metrics of ViT on main dataset.*

Figure A18. *Metrics of ViT on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A19. *Metrics of DeIT on main dataset.*

Figure A20. *Metrics of DeIT on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

Figure A21. *Metrics of BeIT on main dataset.*

Figure A22. *Metrics of BeIT on validation dataset.*

 doi:10.20944/preprints202604.0294.v1

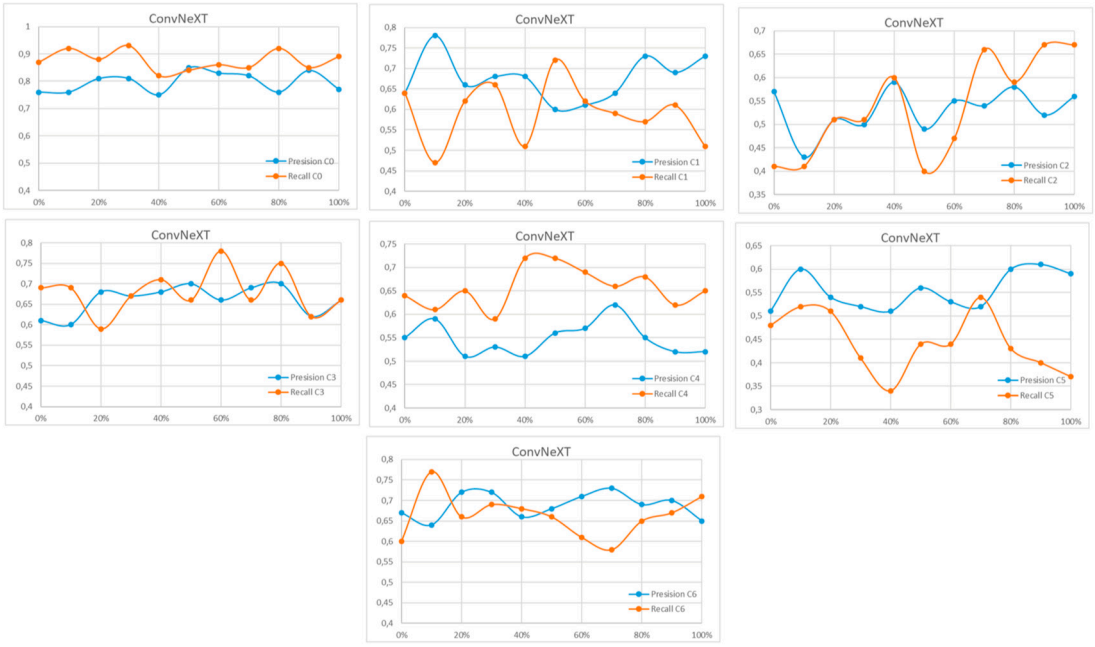

Figure A23. *Metrics of ConvNext on main dataset.*

Figure A24. *Metrics of ConvNext on validation dataset.*

References

1. Yang, S.; Xiao, W.; Zhang, M.; Guo, S.; Zhao, J.; Shen, F. Image Data Augmentation for Deep Learning: A Survey. arXiv 2022, arXiv:2204.08610.
2. Alomar, K.; Aysel, H.I.; Cai, X. Data Augmentation in Classification and Segmentation: A Survey and New Strategies. J. Imaging 2023, 9, 46.
3. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. J. Big Data 2019, 6, 60.

 doi:10.20944/preprints202604.0294.v1

4. Min, F.; Yu, T.; Zhang, C.; Xiao, Y.; Zhang, Z.; Chen, Y.; Xu, C.; Cai, J.; Chen, X.; Li, Z.; et al. A review of medical image data augmentation techniques for deep learning in healthcare. J. Med. Radiat. Sci. 2022, 69, 185–197.
5. Antoniou, D.; Storkey, A.; Edwards, H. BAGAN: Data Augmentation with Balancing GAN. arXiv 2018, arXiv:1803.09655.arxiv
6. Wei, J.; Zou, C.; Coulon, J.A.; Klawonn, F.; Hammer, P.L. Text data augmentation for deep learning. J. Big Data 2021, 8, 134.
7. Alyasin, E.I.; Ata, O.; Mohammedqasim, H.; Mohammedqasem, R. Optimizing Prediction of Cardiac Conditions Using Hyperparameter Optimization and Ensemble Learning. Cogn. Comput. 2023, 15, 1–15.
8. Barulina, M.; Sanbaev, A.; Okunkov, S. Deep Learning Approaches to Automatic Chronic Venous Disease Severity Assessment from Photographs. Mathematics 2022, 10, 3571.
9. Kaggle Skin Dataset. Available online: https://www.kaggle.com/datasets/ahmedxc4/skin-ds (accessed on 29 March 2026).
10. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
11. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv 2021, arXiv:2010.11929.
12. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
13. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Going Deeper With Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
14. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
15. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A ConvNet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022.