Empirical verification of the suggested hyperparameters for data augmentation using the fast.ai library

Wojciech Oronowicz-Jaśkowiak
Faculty of Computer Science of the Polish-Japanese Academy of Information Technology
Koszykowa 86
Poland, Warsaw 02-008
oronowiczjaskowiak@pjwstk.edu.pl

Piotr Wasilewski
Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw
Banacha 2
Poland, Warsaw 02-008
piotr@mimuw.edu.pl

Mirosław Kowaluk
Faculty of Mathematics, Informatics and Mechanics of the University of Warsaw
Banacha 2
Poland, Warsaw 02-008
kowaluk@mimuw.edu.pl

ABSTRACT
Data augmentation consists in adding slightly modified copies of the existing data to the training set, which increases the total amount of data and generally results in better results obtained by machine learning algorithms. The fast.ai library has some predefined values for data augmentation hyperparameters for visual data. It is claimed that these predefined parameters are to be the best for most data types, however, no empirical support for this statement has been provided. The aim of this research is to determine whether the suggested hyperparameter values for data augmentation in the fast.ai library are indeed optimal for the highest accuracy for image classification tasks. In order to answer this question, a detailed research was conducted, consisting of a series of experiments for subsequent data augmentation tools (rotation, magnification, contrast change, etc.). Three variables were modified for each tool: 1. maximal and minimal value of transformation (depending on the transformation type), 2. probability of the transformation, 3. padding behaviour. The results of the presented research lead to the conclusion that the suggested values of data augmentation implemented in the fast.ai library provides the good parameters of the model aimed at differentiating male and female faces, however in case of that classification slightly different parameters could be taken into consideration. The results are published in open-source repository (*Open Science Framework*, DOI:10.17605/OSF.IO/38UJG).

Keywords
Machine learning, supervised learning, computer vision

1. INTRODUCTION
Currently, machine learning uses many libraries to support the process of building advanced models. Libraries and their extensions enable the construction of known algorithms without the need to develop them from the very beginning and ensure flexibility of calculations. Fast.ai library used in this research will be characterized – fast.ai library, for which parameters the experiments were carried out.

The fast.ai library is an open source programming library that, like PyTorch, uses the Python [How00a] programming language. It is highly integrated with the PyTorch library. This library gained publicity at the end of 2018, when a group of students using fast.ai manager to defeat commercial teams from the Google and Intel teams in the CIFAR-10 classification competition. The results of this competition showed that even smaller development teams are still able to present better technological solutions than large corporations with an incomparably larger budget. The fast.ai library is mainly based on PyTorch library, but there are some significant differences compared to it. Fast.ai library allows to easily use very advanced techniques supporting machine learning, such as searching for the optimal *learning rate* range that can be used in the network training process. The documentation for fast.ai library, version 2.0, is extensive and provides a number of examples to illustrate the use of the built-in functions. The advantage of this library is therefore the possibility of quick use of functionalities supporting the process of training neural networks, which from the programmer's perspective are implemented as one line of code, but in fact are much more complex.

2. RELATED WORK
The fast.ai library has been used successively in many research. Recently, the author of the library, Jeremy Howard, proposed the ULMFiT (*Universal Language Model Fine-Tuning*) model based on the

fast.ai library, which allowed to improve the task of text classification on the example of several popular training data sets [How01a]. The technique of audio spectrogram language identification is also presented (LIFAS), using spectrograms generated from audio signals, and then transmitted as input data to the convolutional neural network, which in turn allows for language identification. The technique presented by the authors, thanks to the use of the fast.ai library, is transparent and seems to be easily available for replication [Rev00a]. An interesting article that has been published recently is the proposed classification of malware on the basis of image analysis [Bho00a]. Although the results presented by the authors showed that deep neural networks are not better at this task compared to simpler methods, such as the k-means method, the results related to neural networks may be characterized by a better generalization ability. In both cases (the use of neural networks as well as the k-means method), an analysis was performed using the fast.ai library.

The presented variety of applications shows the great possibilities and flexibility of the fast.ai library. The creators themselves encourage even users who are just starting their adventure with machine learning to experiment on data based on domain knowledge. The idea behind the creators was a kind of democratization of advanced machine learning methods, translating into the development of knowledge in many areas.

Data augmentation consists in introducing such modifications to photos that allow them to be enriched with features that they did not have before [Bho00a]. Thanks to this, important effects can be obtained from the point of view of the parameters of machine learning models. Increasing the amount of training data by introducing slightly modified copies of them helps to counteract rapid overtraining of neural networks, especially when one have a relatively small set of training data. It has been repeatedly indicated that the use of various data augmentation techniques may allow for significantly higher parameters of machine learning models [Zho00a, Won00a, Mas00a].

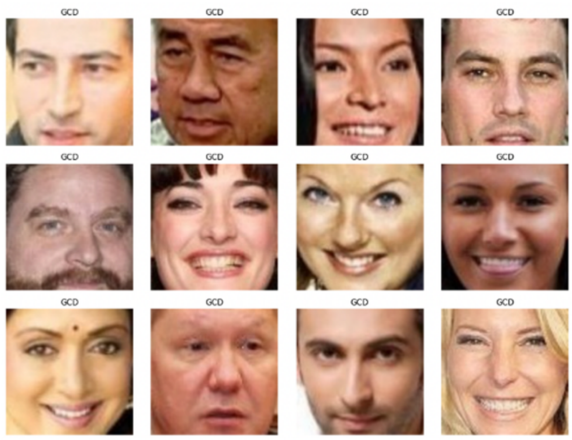
The fast.ai library allows to use several predefined transformations of training data, data augmentation techniques. Among the transformations that are made automatically by calling the *get_transforms* class, the following are distinguished: reflection of the photo in relation to the x and y axes, rotation of the photo by a random angle, zooming, brightening and *warp*.

The aim of the study was to verify the suggested values [How00a] of data augmentation transformations for the fast.ai library. In order to select the best hyperparameters, a total of 26 studies were conducted, which were related to individual techniques of data augmentation, and an initial study related to the selection of the appropriate architecture of the neural network.

3. METHOD

The *Gender Classification Dataset* [Cha00a] was used for the research. This database includes photos of human faces – women and men – of different ages and ethnic origins. These photos are a changed version of the material from another IMDB-WIKI [Cha00a] data set, in such a way that only the images of the face were separated from the entire human figure. The database created in this way contains photos in .jpg format with a resolution of 80x100 px., assigned to one of two categories – photos of female and male faces. In each case, the fast.ai library was used to train the neural networks. The training material consisted of 16,000 photos, 8,000 female faces and 8,000 male faces, respectively. The validation material consisted of 4,000 photos (with a similar division). Two classes were defined – defined as men and women. The training and validation material was selected each time at random according to the relevant research conditions. A total of 100 epochs (network training cycles) were run in ten stages. Each stage consisted of ten eras. If the network overtraining was observed at the end of a given stage, the stage that preceded the overtraining was selected. The Figure 1 shows the examples of training set [Cha00a].

Figure 1. Examples of images used in the training set.



To validate our study, we used four different datasets: CIFAR-10 [Kri00a], Intel Image Classification [Ban00a], sexACT 0.5 [Oro00a] and MNIST [LeC00a]. The first dataset consists of 60.000 color images categorized in 10 classes (e.g. bird, cat, dog, frog). The second dataset consists of 25.000 color images categorized in 6 classes (e.g. buildings, sea, street). The third dataset consists of

11.600 color images categorized in 11 classes presenting human sexual activity (e.g. BDSM). The last dataset consists of 60.000 images categorized in 10 classes (numbers ranged from 0 to 9).

The research that was performed included conducting a series of experiments in which three variables were modified that affect the nature of the applied data augmentation: maximum and minimum size of transformations, probability of making this type of transformations, padding.

Before conducting the appropriate series of experiments, a study was conducted to identify the architecture that would be most appropriate for the selected data type. It was decided to choose one of the few most common architectures in the scientific literature [Cha00a, Sim00a] – ResNet50, ResNet101, ResNet152, VGG16, VGG19. VGG (Visual Geometric Group) architecture is characterised by grouping convolution layers with small kernel sizes, however it is not resistant to explosion of gradients problem. ResNet is the residual neural network presented to solve problem with vanishing gradient. Using that architecture, it is possible to train network with large number of layers.

A total of 100 epochs (network training cycles) were performed for ten conditions – for five selected architectures (ResNet152, ResNet101, ResNet50, VGG19, VGG16) and for two conditions (no data augmentation and the use of suggested values for data augmentation from the fast.ai libraries). The criterion for the selection of the neural network architecture, which was used for later studies, was the maximum value of the classification accuracy on the validation material before the signs of network overtraining.

4. RESULTS

Table 1 and Table 2 show the maximum values of the classification accuracy for individual network architectures.

Table 1. Maximum values of classification accuracy without data augmentation for individual architectures before signs of neural network overtraining.

Research id	Architecture	Accuracy	F-score	Training loss	Validation loss
1.1.	ResNet152	96.40%	0,9674	0.2797	0.1174
1.2.	ResNet101	96.32%	0,9655	0.2854	0.1148
1.3.	ResNet50	96.87%	0,9899	0.2452	0.1275
1.4.	VGG16	96.42%	0,9644	0.2695	0.1246
1.5.	VGG19	96.50%	0,9670	0.2684	0.1455

Table 2. Maximum values of classification accuracy using data augmentation (fast.ai library suggested values) for individual architectures before overtraining.

id	Architecture	Accuracy	F-score	Training loss	Validation loss
2.1.	ResNet152	96.55%	0,9699	0.2721	0.1069
2.2.	ResNet101	96.95%	0,9720	0.2612	0.1183
2.3.	ResNet50	96.85%	0,9684	0.2705	0.1155
2.4.	VGG16	96.65%	0,9671	0.3129	0.1224
2.5.	VGG19	96.67%	0,9664	0.2918	0.1218

As a result of the research, the ResNet101 architecture was selected. Compared to other architecture, ResNet101 had the most stable learning process, the highest final classification accuracy was achieved, the validation loss was relatively low and the highest F-score. Six studies were conducted, where the hyperparameters appropriate for the right data transformations were modified – testing the values of transformations of the class flip_vert (true or false), flip (true or false), max_lighting (from 0.1 to 0.5), max_rotate (from 8.0 to 12.0), max_zoom (from 1.05 to 1.25). Tables 3 and 4 present the parameters of the neural network in the presence of the flip_vert and flip class transformations. Tables 5, 6 and 7 present the parameters of the neural network in the case of the best parameter from the range presented above.

Table 3. Neural network parameters in the case of the presence of the flip_vert class transformation with the method of filling missing zeroes pixels.

Research id	Data augmentation	Validation loss	Train loss	F-score	Accuracy
3.1.	No	0.1180	0.2977	0,9688	96.45%
3.2.	Yes	0.1178	0.3198	0,9649	96.45%

Table 4: Neural network parameters in the case of the presence of the *flip* class transformation with the method of filling in the missing *zeros* pixels.

Research id	Data augmentation	Validation loss	Training loss	F-score	Accuracy
4.1.	No	0.1180	0.2873	0,9677	96.75%
4.2.	Yes	0.1216	0.2858	0,9678	96.72%

Table 5: Neural network parameters in the case of the presence of the *max_ lighting = 0.30* class, with the method of filling the missing *border* pixels.

Research id	Probability	Validation loss	Training loss	F-score	Accuracy
5.1.	0.1	0.1118	0.3047	0,9653	96.62%
5.2	0.2	0.1123	0.2801	0,9655	96.67%
5.3	0.3	0.1034	0.2997	0,9669	96.67%
5.4	0.4	0.1077	0.2964	0,9688	96.65%
5.5.	0.5	0.1134	0.3038	0,9644	96.42%
5.6.	0.6	0.1121	0.2703	0,9710	96.80%
5.7.	0.7	0.1051	0.2704	0,9788	97.15%
5.8.	0.8	0.1094	0.2816	0,9710	96.67%
5.9.	0.9	0.1129	0.2906	0,9699	96.87%
5.10.	1.0	0.1117	0.2910	0,9661	96.60%

Table 6: Neural network parameters in the case of the presence of a *max_rotate = 9.0* class transformation, with the method of filling the missing *border* pixels.

Research id	Probabili ty	Validatio n loss	Training loss	F-score	Accurac y
6.1.	0.1	0.1072	0.2798	0,9656	96.85%
6.2.	0.2	0.1067	0.2763	0,9721	97.00%
6.3.	0.3	0.1085	0.2951	0,9698	96.52%
6.4.	0.4	0.1092	0.2931	0,9681	96.87%
6.5.	0.5	0.1109	0.2940	0,9622	96.75%
6.6.	0.6	0.1074	0.3004	0,9678	96.72%
6.7.	0.7	0.1118	0.2884	0,9643	96.85%
6.8.	0.8	0.2671	0.0799	0,9821	97.92%
6.9.	0.9	0.3081	0.0916	0,9741	97.37%
6.10.	1.0	0.2929	0.095	0,9799	97.52%

Table 7: Neural network parameters in the case of the presence of the *max_ zoom = 1.25* class, with the method of filling the missing *border* pixels.

Research id	Probabili ty	Validatio n loss	Training loss	F-score	Accurac y
7.1.	0.1	0.1111	0.2889	0,9677	96.85%
7.2.	0.2	0.1106	0.2617	0,9688	96.82%
7.3.	0.3	0.1021	0.2924	0,9689	97.17%
7.4.	0.4	0.1108	0.2979	0,9689	96.77%
7.5.	0.5	0.1050	0.2857	0,9699	97.00%
7.6.	0.6	0.1014	0.2884	0,9749	97.32%
7.7.	0.7	0.1071	0.2913	0,9613	96.87%
7.8.	0.8	0.1025	0.2880	0,9650	96.92%
7.9.	0.9	0.1050	0.2937	0,9676	97.05%
7.10.	1.0	0.1091	0.2829	0,9641	96.92%

After selecting the best parameters (flip_vert = False, do_flip = False, max_lighting = 0.3, p_ lighting = 0.70, max_rotate = 9.0, max_zoom = 1.25, p_ affine = 0.60) we conducted additional experiments aimed to compare the results using different datasets. Table 8 shows the maximum values of the classification accuracy for individual classification task.

Table 8. Neural network parameters using different datasets.

Research id	Dataset	Neural network model	Accuracy	F-score
8.1.	CIFAR-10 [Kri00a]	ResNet101, default data augmentation hyperparameters	83,68%	0,8730
8.2.		ResNet101, new data augmentation hyperparameters	83,89%	0,8392
8.3.	Intel Image Classification [Ban00a	ResNet101, default data augmentation hyperparameters	94,01%	0,9421
8.4.		ResNet101, new data augmentation hyperparameters	94,19%	0,9425
8.5.	sexACT 0.5 [Oro00a]	ResNet101, default data augmentation hyperparameters	95,45%	0,9549
8.6.		ResNet101, new data augmentation hyperparameters	95,98%	0,9621
8.7.	MNIST [LeC00a]	ResNet101, default data augmentation hyperparameters	96,02%	0,9612
8.8.		ResNet101, new data augmentation hyperparameters	95,98%	0,9610

5. DISCUSSION

In principle, each of the architectures selected would ultimately allow the differentiation between the data with high precision. There were slight differences between the architectures in terms of the amount of the loss of validation or the final accuracy of the classification. The lowest value of the validation loss in the case of non-augmented data was found in the ResNet101 architecture, while in the case of augmented data it was the ResNet152 network. The most stable course of the learning process, understood as the lack of clear differences in the validation loss and signs of network overtraining, occurred in the case of the VGG19 and VGG16 architectures.

It seems that random rotation of photos showing faces in the task of differentiating between faces of women and men does not translate into obtaining better model parameters. If so, these results would be in line with those proposed by the authors of the fast.ai library, as suggested. The random use of the "mirror image" in the task of differentiating between the faces of women and men does not translate into obtaining better parameters of the model. If so, these results would not be consistent with those proposed by the authors of the fast.ai library as suggested. The suggested value for this transformation is a logical value indicating that this transformation should be performed (*do_flip = True*), while the results of the conducted research indicate that this operation does not lead to better parameters, therefore there is no basis for its use in this set data. In the case of this class of transformations, consisting in random brightening of photos, it was found that the *max_lighting* technique significantly influenced the parameters of the model. The best results were achieved with the transformation class value set to 0.3, with the transformation probability of 0.7., the systematic use of the transformation consisting in changes in the angle of inclination of the photos in the task of differentiating between the faces of women and men translates into obtaining better model parameters. The observed results are not consistent with those proposed by the authors of the fast.ai library as suggested. In the case of this class of transformations, consisting in the random enlargement of images, it was found that the max_zoom technique significantly influenced the parameters of the model. The best results were obtained when the value of this transformation class was set to 1.25, with the transformation probability of 0.6. With such selected parameters, a low validation loss was obtained, amounting to 0.1014, and the highest accuracy of the test material classification, amounting to 97.32%.

It is worth highlighting the limitations of the study. Firstly, the presented study is limited to the images showing faces and only one dataset - *The Gender Classification Dataset* [Cha00a]. It is known that data transformation is different when it comes to train different data. We have tested new parameters using four different datasets, however the results were the best when there were faces in the photos (e.g. in sexACT dataset). This may prove that the new parameters are specific to the face classification task. Secondly, the presented differences in model parameters are slight, consequently they might be related with computational bias. Thirdly, we used only geometric transformations, but there were presented other data augmentation algorithms. Fourthly, to easier test our hypothesis we used only the basic metrics, however to draw more advanced results, different metrics such as NPV, FNR, FDR or ROC curve should be also compared.

6. CONCLUSIONS

The results of the presented research lead to the conclusion that the suggested values of data augmentation implemented in the fast.ai library provides the good parameters of the model aimed at differentiating male and female faces, however in case of that classification slightly different parameters could be taken into consideration. In the case of tests carried out on a selected set of training data:

- A flip_vert class transformation should not be applied (flip_vert = False) and the suggested value of this transformation (flip_vert = False) is consistent with what was observed.
- The flip class transformation should not be applied (do_flip = False) and the suggested value of this transformation (do_flip = True) does not match the observations.
- Transformation of the max_lighting class should be applied (max_lighting = 0.3, p_ lighting = 0.70) and the suggested value of this transformation (max_lighting = 0.2, p_ lighting = 0.75) does not match the observations.
- Transformation of the max_rotate class should be used (max_rotate = 9.0, p_ affine = 1.00) and the suggested value of this transformation (max_rotate = 10.0, p_ affine = 0.75) does not match the observations.
- Transformation of the max_zoom class should be used (max_zoom = 1.25, p_ affine = 0.60) and the suggested value of this transformation (max_zoom = 1.10, p_ affine = 0.75) does not match the observations.

7. ACKNOWLEDGMENTS

8. REFERENCES

[Bho00a] Bhodia, N., et al. Transfer learning for image-based malware classification. ArXiv preprint arXiv:1903.11551, 2019.

[Ban00a] Banslal, P. Intel Image Classification. Accessed 28.05.2022 from www.kaggle.com.

[Cha00a] Chauhan, A. Gender Classification Dataset. Accessed 22.12.2021 from www.kaggle.com/cashutosh/gender-classification-dataset.

[He00a] He, K., et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[How00a] Howard, J. Fastai V2 documentation. Accessed 22.12.2021 from http://docs.fast.ai.

[How01a] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. Accessed 22.12. 2021 from https://arxiv.org/abs/1801.06146; 16.04.2021.

[Kri00a] Krizhevsky, A, et. al. Learning multiple layers of features from tiny images. Accessed 28.05.2022 from: https://www.cs.toronto.edu.

[LeC00a] LeCun, Y. The MNIST database of handwritten digits. Accessed 28.05.2022 from www.yann.lecun. com/exdb/mnist.

[Mas00a] Masi, I., et al. Do we really need to collect millions of faces for effective face recognition?" European conference on computer vision, Springer, Cham, 2016.

[Oro00a] Oronowicz-Jaśkowiak, W, et. al. Binary classification of pornographic and non-pornographic materials using the sAI 0.4 model and the modified database. *Advances in Psychiatry and Neurology*, 29(2), pp.108-119, 2020.

[Rev00a] Revay, S., Teschke, M. Multiclass language identification using deep learning on spectral images of audio signals. ArXiv preprint arXiv:1905.04348, 2019.

[Sho00a] Shorten, C., Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *Journal of Big Data,* 6.1, pp. 1-48, 2019.

[Sim00a] Simonyan, K., and Zisserman, A. Very deep convolutional networks for large-scale image recognition. ArXiv preprint arXiv:1409.1556, 2014.

[Won00a] Wong, S.C., et al. Understanding data augmentation for classification: when to warp? 2016 international conference on digital image computing: techniques and applications (DICTA), IEEE, 2016.

[Zho00a] Zhong, Z, et al. Random erasing data augmentation Proceedings of the AAAI Conference on Artificial Intelligence, 34, 2020.