UDC: 519.8 https://doi.org/10.15407/jai2023.01.037

CLASSIFICATION OF TEXT AS IMAGES USING NEURAL NETWORKS PRE-TRAINED ON THE IMAGENET DATASET

V. Slyusar
Central Research Institute of Armaments and Military Equipment of Armed Forces of Ukraine
Povytrophlotsky Ave, 28B, Kyiv, 03049
https://orcid.org/0000-0002-2912-3149

Abstract. The article proposes a new approach to solving text classification tasks using pre-trained convolutional neural networks for image processing. A comparison of the training results of different neural network architectures was performed for the dataset of text reviews about the Tesla electric car. The obtained results allowed us to conclude that among the analyzed variants of text dataset preliminary preparation, the bag of words (BoW) method provides the best classification accuracy results on average. When using the EfficientNetB0 neural network previously trained on the ImageNet dataset, this approach allowed to obtain an average class accuracy of texts classification of 99.5%. The embedding procedure is somewhat inferior to the BoW method. However, if the proposed variant of data augmentation based on an additional Embedding layer is applied, it can give a more advantageous result for some neural networks. In particular, the neural structure based on Xception in this case made it possible to achieve an accuracy of 98.9%, which slightly exceeded the accuracy recorded for a similar neural network on the BoW dataset (98.4%). The Word2vec method turned out to be the least successful option for text digitization, although it is possible that its significant loss in accuracy can be reduced with a better choice of text vectorization parameters. The proposed approach regarding the combination of the BoW text dataset preparation method with the additional Embedding procedure as part of the neural network deserves attention. Such a combination in the case of EfficientNetB0 made it possible to achieve a relatively high accuracy of 98.7%, which gives reasons to recommend the use of this combination as one of the options that should be tested at the stage of choosing the best neural network architecture.

Keywords: neural network, dataset, ImageNet, text classification, bag of words, Word2vec, embedding.

Introduction

The spread of the use of artificial intelligence technologies in various spheres of social life, among other things, makes the task of automatic text processing relevant. On this basis, various voice services, chatbots, feedback classification, resumes, etc. have been implemented. Successes in the relevant field are primarily related to the development of natural language processing (NLP) methods and their integration with the capabilities of neural networks.

Despite the certain separation of NLP as an independent direction in the development of artificial intelligence, recently there has been a close relationship between NLP and image and video processing. An example of this is the synthesis of images based on text descriptions or the reverse procedure for forming text comments on photographs. In the military sphere, in a similar way, the corresponding operational situation can be formed on the digital map based on the text of the combat order, or the changes made in the locations of symbolic images on the map can be transformed into separate text sections of the corresponding combat order. Transforming texts into images opens up new opportunities for technical design, finding new looks for technical samples, architectural solutions, etc.

At the same time, the specified process of integration of NLP and image processing technologies in neural networks is at the initial stage of its development and has not yet managed to cover all possible directions of relevant interaction. In particular, we are talking about borrowing for text processing well-known solutions for neural network image processing. In this way, tasks of classification of texts, their segmentation, etc. can be solved. However, the corresponding direction has not yet received proper development and needs comprehensive attention.

Analysis of recent research and publications

One of the first approaches to processing texts as images was proposed, for example, in [1]. In this publication, the image was formed as a projection of the array obtained from the output of a pre-trained neural network for BERT text processing. However, the authors of [1] did not

comprehensively consider the possibilities of using previously trained neural networks for image processing together with BERT and limited themselves to only one approach to transforming text into images.

Taking into account the above, the purpose of the article is to consider the possibilities of classifying texts as images using neural networks previously trained on the ImageNet dataset.

It is quite obvious that the key stage of appropriate processing of texts is their transformation into images. This difficult, at first glance, task is actually quite easily solved thanks to the well-known methods of transforming texts into digital sequences.

In particular, in the context of text classification, it is proposed to convert the text into a set of vectors using the sliding window tokenization procedure. In this case, the length of the window is used as the size of the image frame, and the windows themselves ensure the formation of the frame lines. From such a set of vectors, a matrix or tensor is formed, which formally fulfills the role of a traditional image.

To expand the dynamic range of tokenization indices when transforming them into pixels, it is appropriate to use a 10-bit image representation similar to the 10-bit video standard. In this case, token indices can range from 0 to 1023 instead of 0 to 256 as in traditional 8-bit video.

Since it is relevant to use long dictionaries containing more than 1000 words for texts, it is convenient to switch to Full HD, 4K, 8K or even 16K image format.

At the input of the neural network, different arrays of data, which are formed by combining the outputs of the tokenizer, the embedding procedure, etc., can be combined into one image (matrix of indices). It is also possible to combine the output of several tokenizers with different window sliding steps or different dictionary lengths.

For further classification of images obtained from text, neural networks previously trained on the well-known ImageNet dataset should be used [2]. We can consider the effectiveness of the practical implementation of this approach using the example of efficientNetB0 [3], MobilNev3Small [4, 5], Xception [6], Inceptionv3 [7] and VGG16 [8] neural networks. At the same time, as a dataset, we will use a set of reviews on the Tesla electric car, which must be classified into positive and negative. The length of a typical response is 120-150 words. The training sample of the data set contains 2580 positive reviews and 1692 negative ones, that is, the data set has a certain imbalance in the filling of classes. The corresponding proportion between them was preserved in the test sample, which includes 483 positive and 317 negative reviews.

Presenting main material

The Bag of Words method

At the first stage of research, the dataset format presented in the form of a bag of words (Bag of Words, BoW) was used. At the same time, a one-dimensional array (vector) of data with a dimension of 20000 was fed to the input of the neural network. As a basic neural network, with the efficiency of which all proposed architectures were further compared, a convolutional neural network for image classification was used, the structure of which is shown in Fig. 1. Such an architecture, despite its external simplicity, has 128000922 adjustable parameters. Its key element, which allows you to move from text to image processing, is the Reshape layer, which in this case forms a 200×100×1 tensor. The transition to a three-dimensional data structure allowed us to further apply a Conv2D convolutional layer with 32 filters formed using a weight kernel with a 3x3 structure and a single step of its shift. Next comes the Flatten leveling layer, the outputs of which are connected to the fully connected Dense neural layer with the ReLu activation function. The outputs of 200 Dense neurons are thinned by a Dropout layer in a ratio of 10:1. The Dense output layer traditionally for classification tasks contains a Softmax activation function.

The neural network training process was carried out with a training step of 0.001 and batch 32 using the Adam optimizer. The execution time of 80 epochs was 1.5 min. in the standard mode of connection to Google Colab Pro+ with an A100-SXM4 video card equipped with 40 GB of RAM.

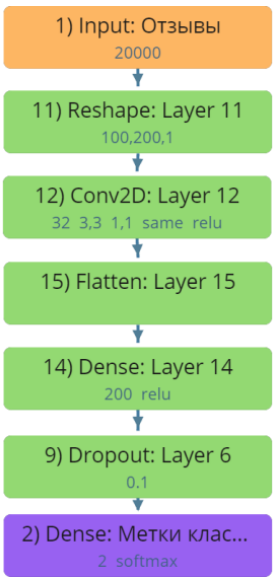


Fig. 1. Basic convolutional neural network for classifying texts as images

During the learning process, the video card resources were used by more than 60% at some stages. In addition to the video card, the Google Colab service also provided 83.48 GB of processor module RAM and 166.77 GB of disk space, but their capabilities were only used at the level of 7 and 15 percent, respectively. The Terra AI framework [9] was used to control the learning process of neural networks in the Google Colab service. At the end of the training on the test sample, the average classification accuracy according to the Balanced Recall metric was achieved at 88.6%. The transition to 64 filters in the Conv2D convolutional layer in combination with the 2x2 weighting kernel made it possible to slightly increase this indicator to the level of 88.7%. In addition, the number of neurons in the first fully connected Dense layer was increased to 256. These changes in the architecture led to an increase in the number of neural network parameters to 327681090.

The principle of BoW text data transformation into an image developed on the basic architecture with the help of the Reshape layer made it possible to move on to the use of pre-trained neural networks. This move dramatically reduced the number of adjustable parameters by ditching the Flatten layer.

In particular, the transition to the use of the EfficientNetB0 neural network (Fig. 2) made it possible to reduce the number of adjustable parameters to 4171638. At the same time, the role of the Conv2D wrapper layer was reduced to the formation of a color image, for which the number of filters in it was set equal to 3. The corresponding conditional image was then subjected to normalization using the BatchNormalization layer. The need for a Flatten layer disappeared because the EfficientNetB0 network in the transfer learning variant has a linear output with a dimension of 1280. In addition, an additional Dense layer with 84 neurons was used instead of the Dropout layer, which provided a smoother transition to the output vector. The specified changes in the architecture of the neural classifier made it possible to increase its accuracy to 91.7%.

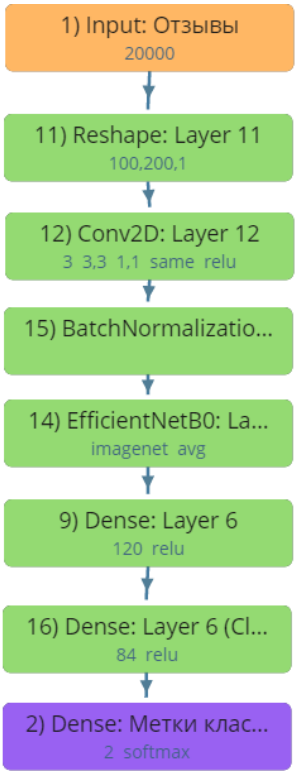


Fig. 2. The proposed neural network for text classification based on the pre-trained EfficientNetB0

Further improvement of this indicator became possible thanks to the method of scaling images at the input of pre-trained neural networks proposed in [10]. For this purpose, the Conv2D layer for the formation of color images was replaced by Conv2DTranspose (Fig. 3) with a 2×4 format kernel and its shift step of 2 and 4, respectively. This made it possible to go from the initial image size of 100×200 pixels to

400×400 format, while the number of neural network parameters increased by 42026, to the value of 4213664.

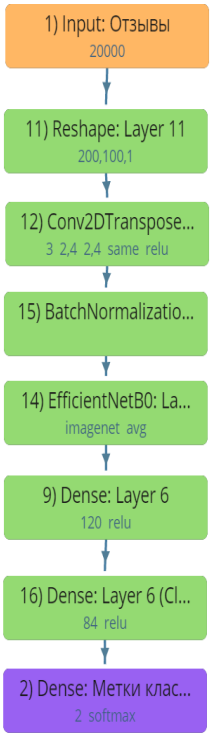


Fig. 3. Modification of the neural network from fig. 2 based on replacing Conv2D with the Conv2DTranspose layer

As a result, the modified neural network made it possible to obtain a classification accuracy of 94.5%. This happened on the 32nd epoch in batch 32 after switching to a training step of 0.00001. All positive responses on the test sample were recognized with 100% accuracy. Unfortunately, when working with the specified modification of the neural network from the Google Colab Pro service, it was possible to obtain computing resources at the level of the TeslaP100-PCIE-16 GB video card. This did not allow to correctly compare the computational cost of training with the basic version and led to an increase in the duration of 40 training epochs to 34.5 minutes.

It should be noted that the choice of parameters of the Conv2DTranspose layer is quite critical not only from the point of view of the effectiveness of training, but also the very possibility of its implementation. In particular, for the same convolution kernel of the 2×4 format, but with a single shift step, the average accuracy on the test selection decreased to 93.4%. Moving to a 3x5 kernel and the same 3x5 shift step allowed us to operate with an image format of 600x500, but with batch 32 this led to the impossibility of training due to exceeding the limited RAM resource. In order to start the process of training the neural network, it was necessary to reduce the batch size to 16, while the training time on the same hardware resources increased to 2 hours. Although the 600x500 frame aspect ratio is better matched to the input image size of the neural network (160/128=1.25; 600/500=1.2) than the square scene, the batch reduction resulted in a previously achieved classification accuracy of 94.5%. However, the kernel and 3×3 shift step parameters in combination with the learning step change strategy surprisingly gave the best result. In particular, the first 20 training epochs were performed with a step of 0.0001, while an accuracy of 87.5% was achieved. After a further transition to a learning step of 0.00001, a sharp jump in accuracy to 96.3% was recorded at the 26th epoch and a record 99.5% at the 27th epoch (Fig. 4). This result proved the success of the chosen approach and the perspective of its application for text processing.

To confirm the non-randomness of the obtained results, the proposed approach to text processing was extended to the use of other pre-trained neural networks. Since most of them have a standard input image size of 160×128 pixels, a decision was made instead of the 200×100×1 format in the Reshape layer to use the 160×125×1 version of the output tensor, the product of whose dimensions is also equal to 20,000.

Among those neural networks that have adaptation to the dimension of the input image, VGG16 was first of all considered.

Fig. 4. Graphs of training accuracy of a neural network based on EfficientNetB0 with a 3x3 Conv2DTranspose layer convolution kernel and the same shift step

In the learning transfer mode (without a "head"), it has a smaller output size compared to EfficientNetB0, which is equal to 512. Experimentally, the limit on the maximum image size at the output of the Conv2DTranspose layer was revealed, which in the case of using VGG-16 was 480×500 pixels and was achieved with 3×4 layer convolution kernel settings and the same shift step. The corresponding version of the neural network architecture is shown in Fig. 5. It has 14786633 parameters, of which only 6 cannot be trained.

As a result of training with a step of 0.0001 already at the 7th epoch, the accuracy of text classification was 96.6%, which fully confirmed all expectations. A slightly smaller image format of 480×375 pixels after the Conv2DTranspose layer (adjusting kernel and shift step 3×3) gave a result of 95.7%. Significantly, when the learning step was increased to 0.001, the response of the neural network to the training process was not observed. This feature is characteristic of most pre-trained neural networks, for training of which it is enough only to fine-tune the pre-set weighting coefficients.

Moving to a simpler neural network MobilNetV3 Small with an output vector containing 1024 elements provided more opportunities to increase the size of the scene using the Conv2DTranspose layer. In fig. 6 shows the corresponding architecture of a neural network operating with images of 800×625 pixels, which are formed in the Conv2DTranspose layer due to convolution parameters and its shift step of 5×5.

When training with a step of 0.0001, it provided a classification accuracy of 92.9% on a test sample of texts. It is significant that such a result was obtained with the connected "head" of the neural network (Include top mode). This approach made it possible to monitor the influence of the image size on the accuracy of the neural network.

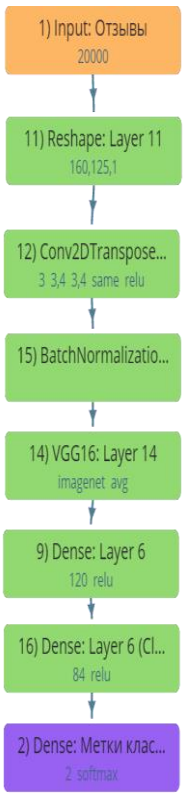


Fig. 5. VGG16 adaptation scheme for text classification

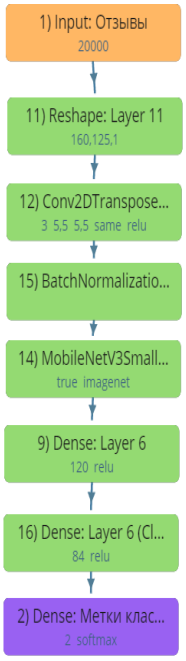


Fig. 6. Neural network architecture based on MobilNetV3 Small

In particular, with the 480×375 format image, the accuracy was limited to 91.1%, and with the 640×500 pixel format, it predictably increased to 92.2%. In general, the simplified architecture of MobilNetV3 Small in this case did not allow to achieve VGG16 and EfficientNetB0 indicators.

Taking into account the experience [10, 11] regarding the effective use of the pre-trained Xception neural network for image classification, the option of using it for text classification was also considered. As you know, a feature of Xception is a fixed input image format of 229×229×3 pixels. Therefore, in order to adapt to this format, the above typical variants of neural networks had to be modified. The main difference was the use of an additionally included Resizing layer set to bilinear mode. This layer transformed the larger format image into a 229×229×3 tensor acceptable for Xception. In addition, to ensure the transition without significant loss of information from the large frame size generated by Conv2DTranspose, an AveragePooling2D layer with the same kernel size and 2×2 pooling step was included before the Resizing block. A variant of the corresponding architecture is shown in fig. 7.

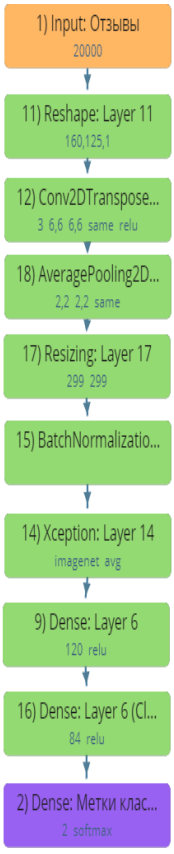


Fig. 7. Neural network of text classification based on Xception

In it, the Conv2DTranspose layer with a convolutional kernel of 6×6 and the same shift interval provided the synthesis of images with a format of 960×750 pixels. At the same

time, the entire architecture had 21117817 parameters, of which 21063283 were subject to training. Training of the obtained neural network with a step of 0.001 at batch 32 made it possible to ensure the accuracy of classification of text responses of 98.4% at the 5th epoch. Further switching to a small training step of 0.0001 did not improve the result, as it led to overtraining.

A structurally similar neural network based on Inceptionv3 with a total number of 22059121 parameters also demonstrated a relatively high accuracy of 95.3%.

Thus, the conducted studies convincingly proved the possibility of effective classification of texts with the help of architectures adapted for image processing. At the same time, the transition from texts to images was carried out in a simpler way than was proposed in [1].

Since the use of the bag of words (BoW) method is only one of the possible options for the digital representation of texts, it is also of interest to consider alternative methods of text digitization, in particular, Word2vec and Embedding.

Word2vec method

According to the Word2vec procedure, the feedback dataset described above was transformed into a two-dimensional array of 100×200 format. Thanks to the application of the Reshape layer, it made it possible to use the architecture of neural networks already considered in previous experiments with almost no changes. In particular, in fig. 8 shows the structure of a neural network adapted to the Word2vec method based on Xception, which differs from fig. 7 only in the input layer.

It should be noted, however, that changing the dataset presentation format to Word2vec led to a sharp deterioration of the classification results. First of all, this is explained by the limitation of the number of vectors chosen to describe the significant words present in the texts. For this reason, some words responsible for the tonality of the response, due to their low frequency of repetition in the texts, were marked as non-essential.

It is also important that the achieved accuracy may vary between different runs of the learning process with the same step. For example, during one of the launches of the neural network presented in fig. 8, at a training step of 0.01, the training did not start even after the completion of 25 epochs, and another time it started already at the 4th epoch. These effects did not only occur in the case of the Xception-based architecture. Regarding it, it should be noted that when the image format after the Conv2DTranspose layer is 600×600 pixels, training with a step of 0.001 gave an accuracy of 71% on the test sample already at the 4th epoch, which was able to be raised to 71.2% at the 45th epoch. Increasing the learning step to 0.01 improved the accuracy to 79%.

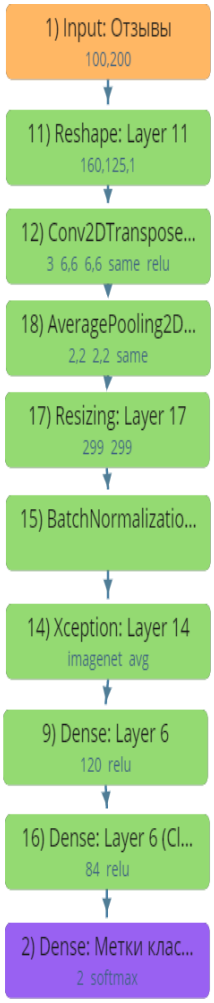
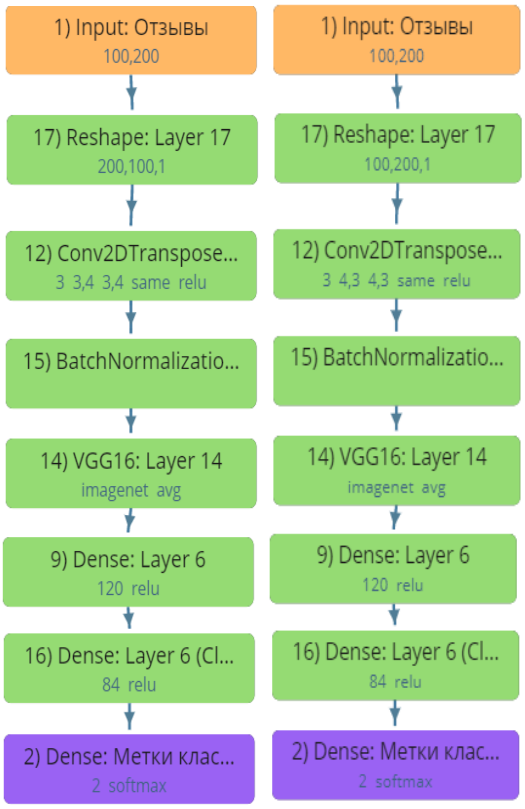


Fig. 8. Modification of the architecture from fig. 7 under the Word2vec method

When reducing the maximum image format to the level of 400×400 pixels (the kernel and the shift step in Conv2DTranspose were 2×2), with batch 32, the maximum

accuracy stabilized at the level of 75.4%. That is, in the case of the Word2vec method, the effect of decreasing accuracy when reducing the image format also occurs.

When using the previously trained neural network VGG16, the fact of its sensitivity to the orientation of the narrow side of the image was recorded. In particular, if the Reshape layer reformatted the input array of data from a 100×200 matrix into a 200×100×1 tensor (Fig. 9.a), the classification accuracy on the test sample was recorded at 71.9%. When Reshape actually duplicated the input image into a 100×200×1 tensor (Fig. 9.b), at a learning step of 0.0001, the neural network did not respond to the training process even after 40 epochs. In both cases, the Conv2DTranspose layer produced a 600×400 image.



а) б)

Fig. 9. Alternative variants of the neural network based on VGG-16

Even greater losses of accuracy were observed when using the EfficientNetB0 neural network. A square data array format of 400×400 elements was created for it in Conv2DTranspose (Fig. 10).

Training began with a step of changing the coefficients of 0.0001, then after 20 epochs, a transition to a step of 0.00001 was made. Such a strategy made it possible to obtain a test accuracy of 63.4%. Replacing the ReLU activation function at the output of EfficientNetB0 with Softmax and starting training immediately with a step of 0.00001 contributed to obtaining an accuracy of 69% at 50 epochs on this architecture.

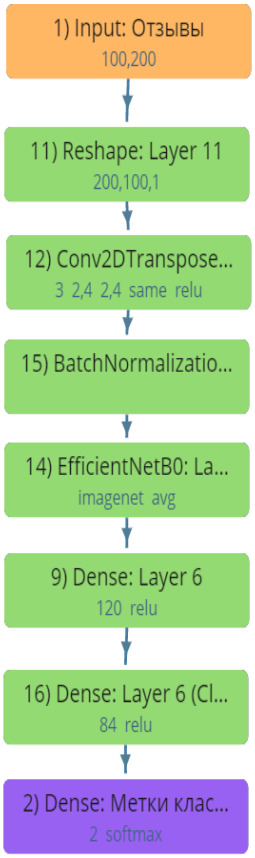


Fig. 10. A neural network based on EfficientNetB0 for the Word2vec method

Embedding method

A feature of the dataset preparation using the Embedding method is the establishment of the maximum number of significant words that will be analyzed in the texts. To maintain continuity with the Word2vec method in question, the specified number of words was limited to 100. At the same time, similarly to the BoW method, the size of the set of words in the texts of the dataset was 20,000.

Another difference between the studied architectures is the connection to the input of the neural network of an additional Embedding layer, for example with parameters 20000×200 (Fig. 11). Re-applying

the Embedding method to text data actually performs the function of augmenting it, given the small size of the input array. At the output of the specified layer, a 100×200 format matrix is formed, i.e. the dimension of the output data is determined by the dimension of the input vector and the maximum number of significant words specified for the Embedding procedure (200). The rest of the structure of the neural network remains unchanged, therefore, for example, for the architecture based on Xception (Fig. 11), the total number of parameters of the neural network has not increased significantly and is 23117799. This indicator remains unchanged when moving from the 200×100×1 tensor to the 160 format in the Reshape layer ×125×1 and fixed settings of the rest of the layers. However, the very result of neural network training depends on the combination of parameters selected in Reshape. With batch 32 with a learning step of 0.001, the use of the 200×100×1 tensor format in Reshape made it possible to obtain an accuracy of 98.9%.

In fig. 12, 13 show the corresponding training results and the error matrix for the test sample.

Since the efficiency of the double embedding procedure as a means of data augmentation was proven by the training results presented in Fig. 12, 13, it was quite appropriate to try to combine it, for example, with the BoW method.

For such an experiment, the EfficientNetB0 neural network previously trained on the ImageNet dataset was taken as a basis. The corresponding architecture is shown in Fig. 14.

In it, the Embedding layer works with a dictionary of 10,000 words, of which 40 are selected as significant (layer parameters 10,000×40). The subsequent Reshape layer forms a 1000×800×1 format tensor, which is then transformed by the Conv2D layer into a conventionally colored image. As a result, a 1000×800×3 format data array is received at the EfficientNetB0 input. The corresponding architecture has 4613688 parameters.

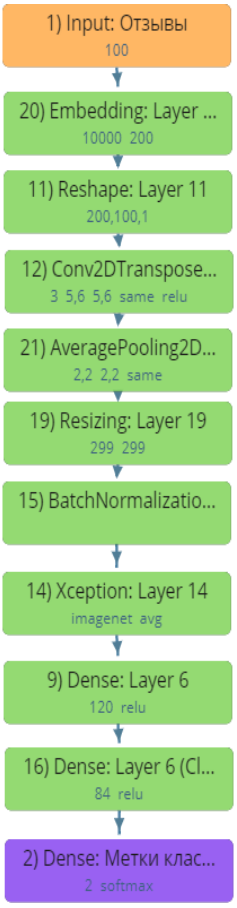


Fig. 11. Scheme of integration of the Xception neural network with the Embedding procedure

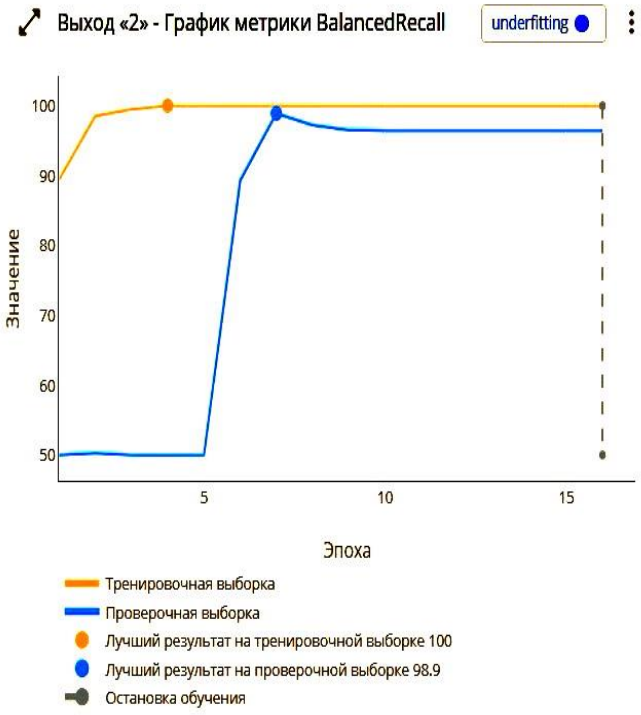


Fig. 12. The results of learning the neural network shown in Fig. 11

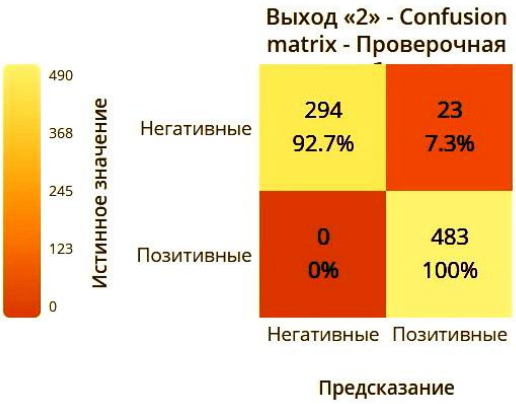


Fig. 13. The matrix of inconsistencies based on the results of the training of the neural network shown in Fig. 11

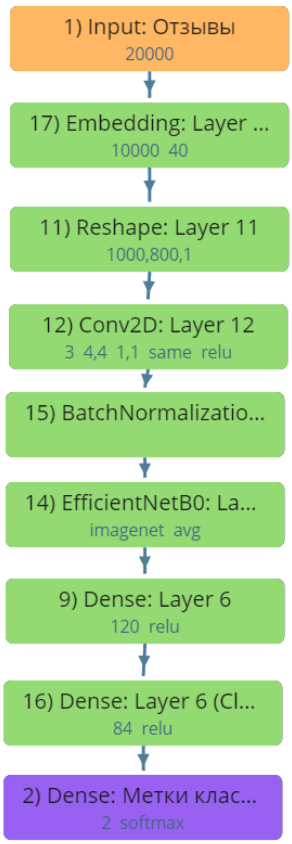


Fig. 14. Option of combining the BoW method with the Embedding procedure

The training of the described structure of the neural network was carried out with a learning step of 0.0001 and batches 4 and 8. With batch 4 at the 6th epoch, an accuracy of 95.9% was achieved, and the use of batch 8 made it possible to raise this indicator to 97.2% at the 18th epoch and 98.7% - on the 25th (Fig. 15).

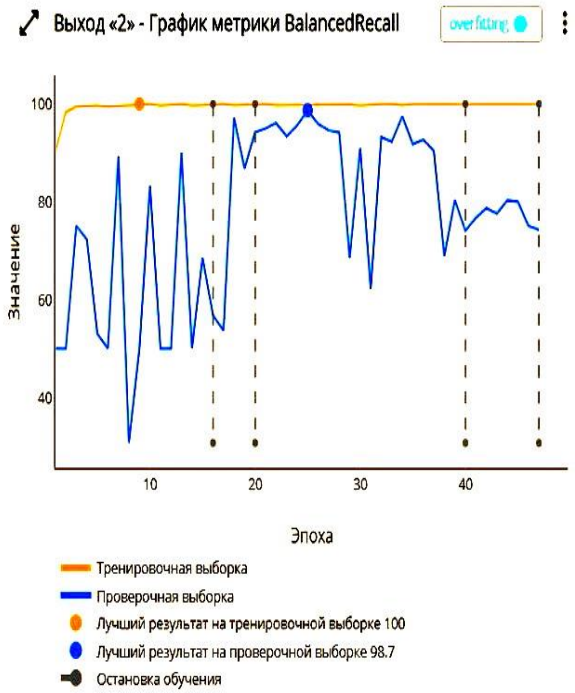


Fig. 15. The results of learning the neural network shown in Fig. 14.

Thus, applying a combination of bag of words and embedding procedure can be one of the effective ways to achieve high accuracy of text classification by image processing methods. However, an attempt to combine the two Embedding layers in one network proved futile, as the corresponding neural structure was unable to learn within 40 epochs at all given learning steps (0.1, 0.01, 0.001, 0.0001, 0.00001).

Conclusions

The conducted studies proved the possibility of effective implementation of text classification tasks with the help of pre-trained convolutional neural networks used for image processing. The approach proposed by the author to the classification of text arrays of data as images is comparatively simpler than that considered in [1]. A comparison of the learning results of different neural network architectures allows us to conclude that among the analyzed variants of preliminary preparation of the text dataset, the best classification accuracy results on average are provided by the bag of words (BoW) method. In particular, when using the EfficientNetB0 neural network previously trained on the ImageNet dataset, this approach made it possible to obtain an average classification accuracy of reviews of 99.5%.

The embedding procedure is somewhat inferior to the BoW method. However, if the proposed variant of data augmentation based on an additional Embedding layer is applied, it can give a more advantageous result for some neural networks. In particular, the neural structure based on Xception in this case made it possible to achieve an accuracy of 98.9%, which slightly exceeded the accuracy recorded for a similar neural network on the BoW dataset (98.4%).

The Word2vec method turned out to be the least successful option for text digitization, although it is possible that its significant loss in accuracy can be reduced with a better choice of text vectorization parameters.

The proposed approach regarding the combination of the BoW text dataset preparation method with the additional Embedding procedure as part of the neural network deserves attention. Such a combination in the case of EfficientNetB0 made it possible to achieve a relatively high accuracy of 98.7%, which gives reasons to recommend the use of this combination as one of the options that should be tested at the stage of choosing the best neural network architecture.

Further research should be focused on extending the scope of application of synthesized neural network structures to solving classification tasks of other variants of texts, in particular resumes, symptoms of diseases, mail messages (spam - not spam), etc. In addition, the proposed approach to identifying texts with images can be considered as an alternative version of neural network technologies for document content segmentation.

References

1. Benarab Charaf Eddine. Classifying Textual Data with pretrained Vision Models through Transfer Learning and Data Transformations. // Feb. 7, 2022, 7 p. arXiv:2106.12479v4. https://arxiv.org/pdf/2106.12479.pdf.
2. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248 – 255. Ieee, 2009.
3. M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. of International Conference on Machine Learning, 2019, pp. 6105-6114.
4. Sandler, M., Howard, A., Zhu, M., et al. (2018) Mobilenetv2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, 18-23 June 2018, 4510-4520. DOI: 10.1109/CVPR.2018.00474.
5. Howard, A., Sandler, M., Chu, G., et al. (2019) Searching for Mobilenetv3. Proceedings of the IEEE International Conference on Computer Vision, Seoul, 27 October-2 November 2019, 1314-1324. DOI: 10.1109/ICCV.2019.00140.
6. F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800-1807, DOI: 10.1109/CVPR.2017.195.
7. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016 pp. 2818-2826. DOI: 10.1109/CVPR.2016.308.
8. H. Qassim, A. Verma and D. Feinzimer (2018), Compressed residual-VGG16 CNN model for big data places image recognition, *Computing and Communication Workshop and Conference (CCWC) 2018 IEEE 8th Annual*, 169-175.
9. Vadym Slyusar, Mykhailo Protsenko, Anton Chernukha, Vasyl Melkin, Olena Petrova, Mikhail Kravtsov, Svitlana Velma, Nataliia Kosenko, Olga Sydorenko, Maksym Sobol. Improving a neural network model for semantic segmentation of images of monitored objects in aerial photographs.// Eastern-European Journal of Enterprise Technologies.- № 6/2 (114). – 2021. - Pp. 86 – 95. DOI: 10.15587/1729-4061.2021.248390.
10. Slyusar V. Architectural and mathematical fundamentals of improvement neural networks for classification of images. // Artificial intelligence, 2022, №1.- Pp. 127 - 138. DOI: 10.15407/jai2022.01.127.
11. Slyusar V.I., Sliusar I.I. (2021) Lions of Neural Networks Zoo, *NeyromerezhnI tehnologIyi ta yih zastosuvannya NMTIZ-2021: zbirnik naukovyh prats XX Mizhnarodnoyi naukovoyi konferentsiyi «Neyromerezhny tehnologii ta yih zastosuvannya NMTIZ-2021»*, Kramatorsk: DDMA, 129 -133, DOI: 10.13140/RG.2.2.17187.58405.

The article has been sent to the editors 31.10.22.
After processing 20.11.22.