Using a Variational Autoencoder in the Task of Overlaying Multiple Images

Gregory Malyshev [1], Vyacheslav Andreev [2], Olga Andreeva [2], Oleg Chistyakov [1] and Dmitriy Sveshnikov [1]

[1] *JSC «Afrikantov OKBM», Burnakovsky passage 15, Nizhny Novgorod, 603074, Russia*
[2] *«Nizhny Novgorod state technical university n.a. R.E. Alekseev», Minina 24, Nizhny Novgorod, 603950, Russia*

Abstract

The possibility and expediency of using variational autoencoders when expanding training datasets of neural networks for cases when the training set consists of several dozen samples is tested. Investigations are carried out on the example of images of «crack»-type defects. Brief information on the theory of variational autoencoders is given. Practical recommendations are given for constructing training sets of variational autoencoders. It is shown that deviation from the suggested recommendations will most likely not allow generating realistic images for the case of a small dataset. For the case of a dataset of eighty images, the distribution of «crack»-type defects in the hidden space after training the variational autoencoder is demonstrated. Examples of images with defects sampled from different parts of the distribution of latent factors are given. For the case of images of cracks, the continuity of the hidden space is demonstrated, when one image is sufficiently smoothly transformed into another on the way through the space of hidden features. A method for obtaining superimposed images based on the use of variational autoencoders is proposed. This method seems to be promising, since it allows automating the process of obtaining superimposed images. Examples of generated cracked superimposed images are shown.

Keywords

neural networks, variational autoencoders, hidden factor space, generative models, defects

1. Introduction

An important task facing the specialists of most manufacturing enterprises is a visual inspection of the surface of products for defects. In order to improve the quality of control of samples and reduce the time of inspection of samples, it is proposed to use a convolutional neural network [1-2], which will make it possible to classify defects on products. One of the main problems that researchers face when training neural networks is the small number of images with defects on which the network will be trained. The emergence of generative adversarial networks [3-6] and variational autoencoders [7-10] naturally leads to the idea of using generator data to expand the training dataset. The purpose of this article is to test the possibility and feasibility of using variational autoencoders to expand small training sets.

2. The concept of a variational autoencoder

An autoencoder is a device that consists of two parts - an encoder and a decoder. The task of the encoder is to downsize input data such as images. For example, we have a black and white image that

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia
EMAIL gsmalyshev@okbm.nnov.ru (G. Malyshev); vyach.andreev@mail.ru (V. Andreev); andreevaov@gmail.com (O. Andreeva); gsmalyshev@okbm.nnov.ru (O. Chistyakov); budnikov@okbm.nnov.ru (D. Sveshnikov);
ORCID: 0000-0002-8147-988X (G. Malyshev); 0000-0002-7557-352X (V. Andreev); 0000-0001-9581-3028 (O. Andreeva); 0000-0002-6515-9691 (O. Chistyakov); 0000-0002-2152-5346 (D. Sveshnikov);

consists of D pixels. Consider this image as a vector in D - dimensional space. The encoder's task is to map a D-dimensional vector into a vector of a lower dimension d. The d-dimensional space of vectors itself is called hidden (latent) or the space of hidden factors. The resulting d-dimensional vector contains information about the most characteristic features of the original images. The task of the decoder is to map the hidden vector into an image that is as close as possible to the original one. Neural networks act as an encoder and decoder.

The main idea of variational autoencoders [7-10] is that the coordinates of the vectors of the hidden space are considered as continuous random variables with their own mean and variance, and the encoder network generates not the hidden vectors themselves, but the parameters of their distributions, that is, the mean and variance [11-12]. Thus, the encoder network must have outputs: d outputs are the mathematical expectations of the latent vector components, the remaining outputs are the standard deviations of the components. The decoder network will have d inputs. The distribution of the hidden space points obtained after training the autoencoder should be close to the normal distribution with a zero mean value vector.

Taking into account the research results presented in the sources [7-8, 13], we can conclude that training images that carry some essential feature (for example, a smile on a person's face) should not differ significantly. For example, if there is a need to construct a hidden vector [7-8] encoding a smile on a human face, then the autoencoder needs to present a set of images of smiling faces that are photographed from a close angle and occupy a fixed location in the photograph. Moreover, no matter what emotions of a person are captured in a photograph, all these images have the same abstract features - nose, eyes, mouth, ears. And these abstract features should not be significantly displaced relative to each other in the photographs from the training set. Otherwise, in the process of training, a certain cluster of hidden space can be allocated, which encodes minor features [14], for example, a background texture. At the same time, an important feature will be "suppressed": instead of identifying an essential feature, the network in the learning process will be "distracted" by processing unnecessary information.

3. Attempting to smoothly sample cracked images in the case of a small training set

As an example, we will consider images of "crack" -type defects. The set consists of 80 black and white images. Examples of some of them are shown in Figure 1. On these images, three essential features can be identified: broken lines (a sign of the crack itself), the surface texture of the product, as well as sharp transitions (product boundaries). For simplicity of visualization, we will consider a two-dimensional hidden space. The challenge is to try to construct latent vectors that encode these essential features. In the process of solving the problem, we will establish how suitable the prepared training set is for training a variational autoencoder.

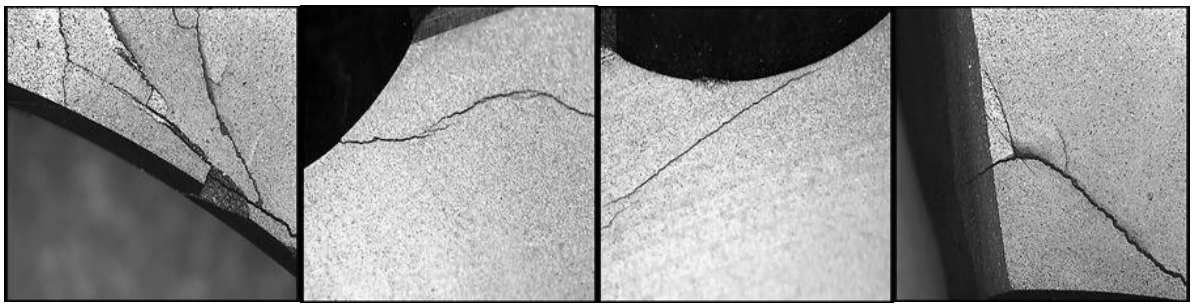

Figure 1: Examples of defects of the «crack» type from the training autoencoder dataset

The variational autoencoder was based on the architecture proposed in the source [15]. The training was carried out for 100 epochs, the size of the minibatch was 10. The RMSprop method was chosen as the optimization method [1]. When training the network using Keras library functions, the initial learning rate was set to $2 \cdot 10^{-3}$. The rest of the parameters of the RMSprop algorithm, as well as the initial values of the weights and thresholds, were left by default.

The representation of the distribution of "crack" -type defects in the hidden space after training is shown in Figure 2.

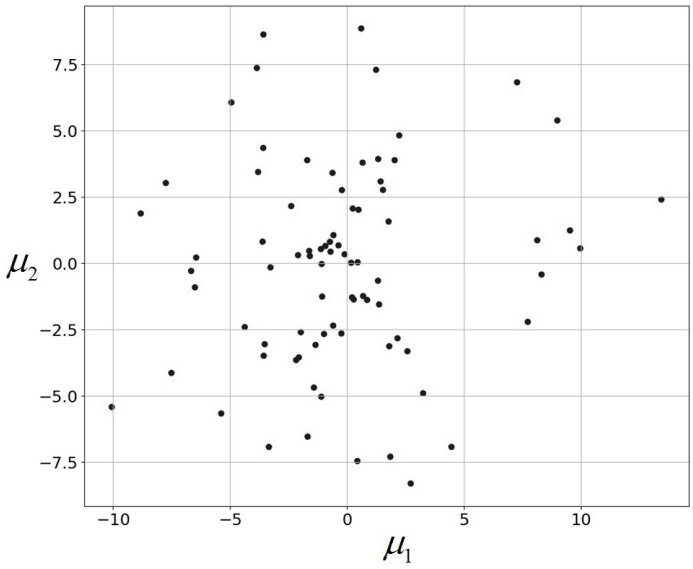

Figure 2: Representation of the distribution of «crack» type defects

The dots in the figure show the coordinates of the $\{\mu_1(x_i), \mu_2(x_i)\}$ vectors of the mathematical expectation obtained for the images x_i from the training set. It can be noted that the obtained distribution really, in the first approximation, resembles a normal distribution with zero expectation. In the resulting distribution, it is difficult to identify areas encoding one of the three essential features. First of all, it is important to find the hidden vectors encoding the broken lines. On the Figure 2 it is possible to estimate the area from which the hidden vectors will be selected: to generate artificial images, values from the segment [-15; 15] must be fed to the first input neuron of the decoder, and on the second input neuron the values from the segment [-10; 10] should be chosen. We will move in increments of 0.1 for each segment. Examples of the images obtained are shown in Figure 3.

It is clearly seen that in the learning process, it was possible to effectively encode only one essential feature - abrupt transitions, that is, product boundaries (Figure 1). In addition, an area of hidden space has been identified that encodes cracks in the images. The images in Figure 3 well demonstrate the continuity of the hidden space, when one image is rather smoothly transformed into another on the way through the space of hidden features.

However, all the cracks that can be discerned in the generated images are only a small part of the dataset. It is interesting to note that the cracks in Figure 3 are almost always double, that is, one crack stretches along the other. Perhaps this very structure of defects was understood by the autoencoder as an essential feature. That is, during the training process, the autoencoder was unable to identify a certain universal feature inherent in all cracks in the training set: neural networks "looped" only on a few cracks from the dataset and revealed essential signs of these cracks.

An explanation for this effect was given earlier: when constructing a hidden vector encoding some essential feature, training images carrying this feature should not differ significantly. That is, the training package was initially prepared incorrectly. From the samples shown in Figure 1, it is clearly seen that all broken lines in the images have different orientations, geometrical sizes and structures. In addition, abstract features cannot be identified for cracks (such are, for example, lips on a person's face), so the requirement that abstract features should not be significantly displaced relative to each other in photographs from the training set loses its correctness.

Figure 3 below shows examples of fractured images sampled from different parts of the latent factor distribution.

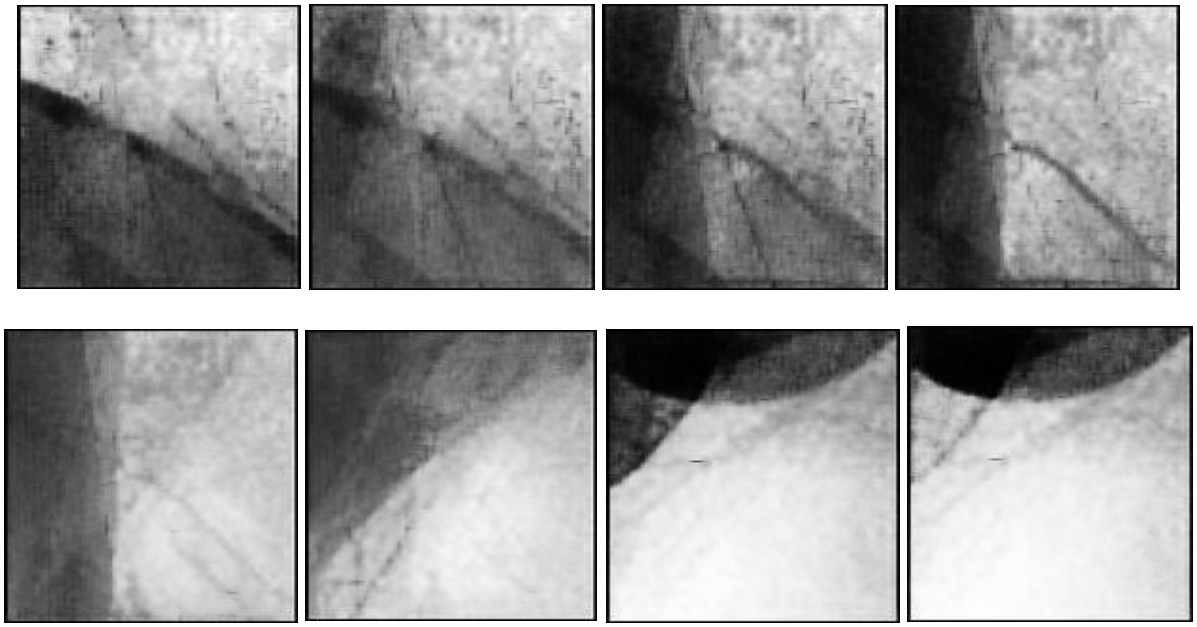

Figure 3: Examples of cracked images, sampled from different parts of the latent factor distribution

The above example showed that the use of a small number of significantly different images, most likely, will not allow significantly expanding the training set using a variational autoencoder, since there are great difficulties in constructing a hidden vector that determines the geometry of the fractures.

4. Image overlay method

It is important to observe the requirement of close proximity of essential features on images from the training set only in cases where there is a small training set, and at the same time it is required to obtain smooth transitions between the generated images. A small set (of the order of a hundred images) of significantly different images can be used to obtain superimposed images, when the generated image is an overlay of several cracked images.

The variational autoencoder in the learning process will construct hidden vectors that encode essential features for different groups of images (for example, for faces and numbers). The more the groups of images differ, the further the indicated vectors will be spaced apart in the hidden space. To obtain a superimposed image, vectors from distant clusters are summed, the resulting vector is fed to the decoder [1-2].

The main idea behind obtaining superimposed images is that essential features must be selected for each image separately. As a result, scattered clusters will be obtained in the hidden space, in which the essential features of each image are encoded (the number of clusters will be equal to the initial number of images). To do this, you need to prepare many copies (at least five hundred) for each image. In the case of ordinary autoencoders, such a technique would be completely useless. For a variational autoencoder, the situation is different: the hidden vectors arriving at the input of the decoder in the learning process are sampled using a "trick" (this is an established term) of reparametrization [7, 16]. The peculiarity of this trick lies in the fact that at each training iteration, new coordinates of the hidden vector will arrive at the input of the decoder, even if the same image is supplied to the input of the encoder.

Let us illustrate the proposed technique (Figure 4, Figure 5).

Figure 4: Sample images on the basis of which the overlay images will be generated

We will restrict ourselves to only three images shown in Figure 4. We will generate an image on which these defects will be combined (two defects of the "crack" type and a defect of the "pore" type). Let's create 500 copies for each of the defects. The architecture of the encoder and decoder was left the same, the dimension of the hidden space is equal to two. Inside the minibatch should be presented samples of those images that we want to combine. The training took place over 20 epochs.

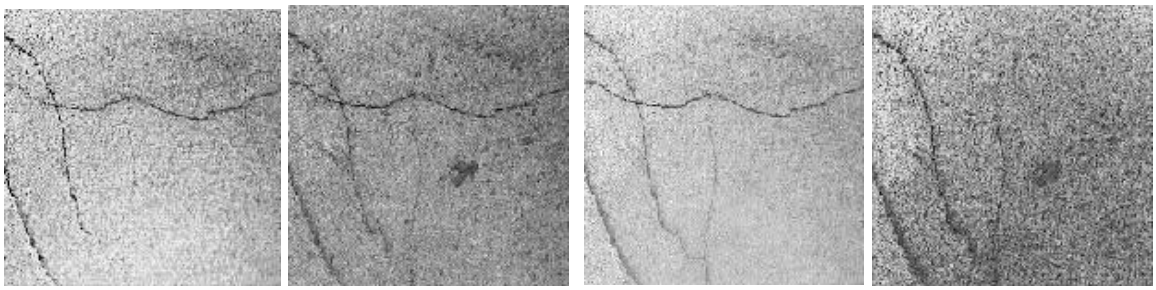

Figure 5: Examples of superimposed images with defects obtained using a variational autoencoder based on the images in Figure 4

The cracks in the generated image (Figure 5) look clearer than they are now, since there were twice as many cracks in the training set of images. The realism of the image can be increased by increasing the dimension of the hidden space. At the next stage of the study, it is necessary to use the obtained overlay images to expand the training dataset intended for training neural networks that perform classification tasks. Comparison of the percentage of correctly recognized images for the cases when the network was trained on the original and extended sets, allows to evaluate the efficiency of using the generated images in the process of training the network.

In [17], a similar problem was solved to quantify the accumulation of damage in metal structures during cyclic loading using a cellular automaton and a neural network that performs the classification task. Within the proposed procedure, images were generated by applying the developed rules of behavior of the cellular automaton. Each iteration of the cellular automaton simulated a load cycle of a metal structure from 1 to N. The cell value on the $N + 1$ cycle was determined by the state of its neighbors from the Moore neighborhood on the N-cycle. The images generated in this way were used to form the training set of the convolutional neural network presented in [18]. The best result in the study was 86.43%. Analysis of the obtained neural network classifier in the work's results suggests that the use of additional images can increase the classification accuracy by 8%. The results of the study [17, 18] allow us to expect a similar effect (increasing the accuracy of the classifier), when the convolutional network with the similar structure is used and trained on a sample of images, supplemented with an autoencoder.

5. References

[1] E. Arkhangelskaya, Deep Learning. Immersion in the world of neural networks, SPb. Peter, St. Peterburg, 2020.

[2] F. Scholle, Deep Learning in Python. SPb. Peter, St. Peterburg, 2020.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in Neural Information Processing Systems, 2014, pp. 139–144. doi: 10.1145/3422622.

[4] I. Goodfellow, Nips 2016 tutorial: Generative adversarial networks, arXiv preprint arXiv:1701.00160, 2016.

[5] S. Nowozin, B. Cseke, R. Tomioka, f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization, in: Proceedings of the 30th International Conference on Neural Information Processing System, NIPS'16, Curran Associates Inc., Red Hook, NY, 2016, pp.271–279. doi: 10.5555/3157096.3157127.

[6] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. Machine Learning, arXiv preprint arXiv:1511.06434, 2015. URL: https://arxiv.org/abs/1511.06434

[7] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, in: Proceedings of the 2nd International Conference on Learning Representations (ICLR), ICLR 2014, Banff, AB, Canada, arXiv:1312.6114.

[8] C. Doersch. Tutorial on variational autoencoders. arXiv:1606.05908, 2016. 3.

[9] P. Goyal Nonparametric Variational Auto-Encoders for Hierarchical Representation Learning in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5104-5112. doi: 10.1109/ICCV.2017.545.

[10] Z. Hu, Y. Zichao, R. Salakhutdinov, E. P. Xing, On Unifying Deep Generative Models, in: Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver Convention Center, Vancouver, BC, Canada, 2018. arXiv:1706.00550

[11] D. J. C. A MacKay, Practical Bayesian Framework for Backpropagation Networks, Neural Computation (1992), 448–472. doi: 10.1162/neco.1992.4.3.448.

[12] R. M. Neal, Bayesian Learning for Neural Networks, Secaucus, NJ, USA: Springer, New York, NY, 1996.

[13] J. H. Metzen, Variational Autoencoder in TensorFlow, 2015. URL: https://jmetzen:github:io/2015-11-27/vae:html.

[14] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, P. Abbeel, Variational Lossy Autoencoder, in: Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Palais des Congrès Neptune, Toulon, France, 2017. arXiv:1611.02731.

[15] Keras Variational AutoEncoder, 2020. URL: https://keras.io/examples/generative/vae/.

[16] L. Chongxuan, Z. Jun, S.Tianlin, Z. Bo, Max-margin deep generative models, in: C. Cortes, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, of NIPS'15, MIT Press, Cambridge MA United States, Montreal Canada, 2015, pp. 1837–1845. URL: https://dl.acm.org/doi/10.5555/2969442.2969445.

[17] A.V. Gonchar, O.V. Andreeva, M.S. Anosov, Study of fatigue failure of construction steels by using modern methods of digital processing of microstructural images, Materials Today: Proceedings of International Conference on Modern Trends in Manufacturing Technologies and Equipment 2020 (ICMTMTE 2020), volume 38, part 4, 2021, pp. 1701-1705. URL: https://doi.org/10.1016/j.matpr.2020.08.226.

[18] O.V. Andreeva, Model and algorithms for damage assessment microstructures metal surfaces and alloys by images, NSTU, Nizhny Novgorod, 2018.