DOI: 10.14529/ctcr170114

THE PROPERTIES OF COMPUTING PROCESSES IN IMAGE ANALYSIS AND MACHINE LEARNING TASKS

***I.V. Parasich**, parasichiv@mail.ru,*
***A.V. Parasich**, parasich_av@yandex.ru*
South Ural State University, Chelyabinsk, Russian Federation

The solving process of any computer vision or machine learning task can be represented in form of some sequence of computational operations on input data. The feature of intelligent data analysis is significant input data heterogeneity which includes emissions, measurement uncertainty, and multimodality. Different types of computing operations respond differently to the presented types of mismatches. The quality of problem solution to a large extent depends on the properties and data mismatch stability of the basic operations. The article describes main types of computational operations, used in computer vision and machine learning algorithms, the analysis of their resistance to various types of mismatch in the data is provided. The information will be useful in designing visual objects descriptors, in the development of detection and tracking algorithms. Of particular value is using of the information in design and analysis of deep convolutional neural networks.

Keywords: computer vision, machine learning, convolutional neural networks, tracking, filtering.

In the process of solving of the computer vision and machine learning tasks, it is often useful to analyze the algorithm from point of view of used computational operations and their properties in order to reveal the problem parts of algorithms and find the ways of increasing the quality of their work. Let us consider the main operations on input data set used in computer vision tasks and the main types of data mismatch that may affect the accuracy of the results. A set of input data in the article refers to a set of pixels within the image window or to an set of classifier votes for the position of an object of interest in the frame in *Implicit Shape Model* [1]. As a set of input data can also be a learning sample of machine learning algorithm, data set to linear regression, set of data of pattern recognition algorithm, etc.

Let us consider the main types of data mismatch and introduce some definitions.

Mismatch – the situation when the real properties of data set are not consistent with the assumptions laid down explicitly or implicitly in the mathematical model used for processing of these data.

Outliers – the kind of mismatch – single data object that has very large or small value because of an error in input data or an error in the previous data processing algorithms.

Imbalance – the kind of mismatch when the mathematical significance of the contribution of the values of some type to the result of the calculation does not correspond to their semantic significance. It has a decisive influence on the result of the algorithm.

Characteristic imbalance – the inconsistency of algorithmic or mathematical significance of a set of values to their semantic significance because of specific influence of some features of these values on operation of processing algorithm. The example of characteristic imbalance is the misrepresentation of proportions of the examples of different classes in the training set in the case of the classifier learning. Let us consider the example of characteristic imbalance. Assume that in using of the algorithm *k-NN* with *k*-1 there are two classes $Y = \{A, B\}$, and *B* class variance is much higher than *A* class variance because of the presence of outliers. The *A* class examples do not cross the class boundary and *B* class examples do cross. On an independent test set, some *A* class examples will be incorrectly determined as *B* class examples because they will be closer to the noise representative of class *B* than to the representative of class *A*. This will reduce the quality of the classification and increase the percent of examples determined as *B* class examples (this percent will be higher than the one in learning sample). Such mismatch is the result of the specific interaction of data items characteristics (noise variance) to a processing algorithm. If the characteristics of *A* class and *B* class representatives are the same the violation of the ratio of the proportions of recognition would not have happened. Characteristic imbalance is difficult to recognize in real tasks.

1. Properties of computer operations

Let us consider directly the types of mismatches for which it is required to ensure robustness during development of data processing algorithm.

Robustness to outliers. Outliers may appear due to errors in measurements (defective pixels in the camera) or because of the errors in the classification algorithm. In some algorithms, great outliers are able to reduce significantly the quality of the working method. Special measures are required to control outliers.

Robustness to measurement errors. There is always a certain amount of error in measurements of observed values and in evaluation of hidden variables. If the accuracy of calculations does matter, one should take measures to compensate such errors.

Robustness to imbalance. While working with value set one cannot guarantee as a rule the ideal from semantic point of view proportion of different types of values. The violation of the ratio of values of different types can be completely arbitrary. For example: one part of object is seen better than the other; one part of object is situated closer to chamber because of what takes more pixels; the window border got more background pixels than object pixels, etc. In machine learning tasks the imbalance of different types of learning examples in learning samples is very common. To eliminate the imbalance the various kinds of normalization are often used. However, such decisions are not always possible; therefore, it is desirable that the proportion of values of different types in data set does not have a decisive influence on the result of the algorithm.

Robustness to small changes of the input data. As a rule on small change of input data algorithm has to react with small change in the result, otherwise there will be experienced the unrobustness. Common example – the multi-frame tracking algorithm, for static pose of object when changing frames the small changes of input data will be observed and the small change of the result of tracking is supposed, the jitter result in such cases is unacceptable. Small changes can be divided into two types: small changes of observed variables (for example, the brightness of the pixels) and small changes of hidden variables (the slight movement of the object in the frame).

Robustness to irrelevant data. When performing calculations on data, not all data elements can refer to objects of interest (the image window can get background or other objects pixels). It is desirable to eliminate or reduce the effect of such elements on result.

Reaction on multimodality. Very often, various elements of processed data set have different nature. This is expressed in the data multimodality. Different cases require different reaction to the presence of multiple modes in the input data. Sometimes it is necessary to take into account equally all modes of distribution. Sometimes it is desirable to give the answer inside one of the modes but not midway between modes (in tracking task). Sometimes false modes should be ignored (when background details and other foreign objects enter the window). In different cases, different operations and computation algorithms appropriate to apply.

Let us analyze the work of the various operations under the conditions of previously discussed mismatches.

Summation, averaging. Unstable to imbalance. By its nature, the result of summing depends on proportion of summands of different types because of what the imbalance critically affects the result. That is why decision trees learning is extremely unstable to imbalance because in the calculation of entropy the summing of numbers of different classes examples is used. Partly unstable to outliers. In the presence of outliers, incorrect values necessarily influence the result. If the outlier is not fundamentally different from the normal data, the large volume of correct data can compensate it. Therefore, if outliers are a small fraction of the total data their influence is not crucial. If the outlier has a value different from the normal at several orders, the total result will be spoiled. Therefore, these outliers should be filtered beforehand. When one summarize a large number of variables the measurement errors cancel each other out if the errors are independent. The more data is added the lower the total error. Stable to small changes in the data. Absolutely unstable to irrelevant data. To operations of such type, the convolution with kernel can be attributed (for example, with Gaussian [2]). In the case of averaging the multimodal data the result can belong to none of the modes (sometimes it is unacceptable) due to which the answer may be obviously wrong, all modes will have an impact on the result. Summation and averaging are used in average-pooling layers of convolutional neural networks.

Weighted summation. Weighted summation goal is to reduce the impact of imbalance and of irrelevant data on the summation result. However, the imbalance may occur if the weights have been chosen incorrectly or if the weights of some elements are obviously more than the weights of the other ones. Weighted summation is used in normal neurons of neural networks (including convolutional neural networks).

Selecting of one element (maximum, minimum, median). Stable to imbalance. Another element may be selected for small data changes. For example, if the cell is selected with the maximum sum of the weights of elements, a completely different cell may be selected for the next frame, that leads to jilter of result. The selecting of median is absolutely stable to outliers. The selecting of minimum and maximum is absolutely unstable to outliers. Absolutely unstable to measurement error (inaccuracy of measurement of the selected value directly becomes inaccurate result), this can be a critical disadvantage in some applications. In the case we operate with multimodal data the result belongs to one mode and the rest modes will not affect the result. If the median is close to the border of modes, the result is unstable to small data changes.

The useful feature of the element selection operation is screening of potentially unnecessary data, a part of erroneous measurements is ignored and does not affect the result, also the robustness to irrelevant data appears. The example of using such type of operation is max-pooling layer [3] in convolutional neural networks. One more classic example – median filter [2] which is used for reduction the outliers.

Threshold condition. The values that are close to threshold are extremely unstable to small changes. Very often, such behavior is unacceptable because it can lead to unrobustness of recognition results and to incorrect work of algorithm in some cases. The use of threshold conditions is often simply necessary. The threshold conditions are contained in any conditional statements without which it is impossible to write a recognition algorithm. In two-class classification task regardless of the algorithm structure, in the end we will have to use the threshold condition for the selection of one of two classes. In particular, decision trees [4] completely consist of a large number of threshold conditions.

One common solution to the problem of unrobustness to small changes is replacement of step threshold function on a smooth one (for example, on sigmoid in neural networks). Another possible solution is majority voting over results of several threshold functions perhaps calculated for different frames (discussed further).

In tracking algorithms in order to prevent constant switching of algorithm from one state to another when control variable fluctuates around the threshold the hysteresis technology is used – reverse switching threshold is done less than direct switching threshold.

Masking. The sum of only those pixels that fall into the mask. If properly chosen the mask increases robustness to irrelevant data. Masking is used in convolutional layers of convolutional neural networks [5].

Taking a cell address. Used for example in the construction of histograms, HOG and SIFT descriptors. As in using the threshold conditions, there is unrobustness on the borders.

Majority voting. From a set of values, the value that occurs most often is chosen. It is used in particular in ensemble of classification algorithms (decision forest). The majority voting used over threshold functions reduces the imbalance to small changes. Practically always, the using of the composition of multiple classifiers instead of single classifier improves the quality of the algorithm largely due to the fact that in case of dispute (near the threshold) the decision is made more flexible, taking into account a large number of data.

2. Analysis

Combined schemes. Summation and averaging operations reduce computing error but are unstable to rough outliers, imbalance and irrelevant data. Operations of selecting one element on the opposite are stable to mentioned above problems but are not able to reduce the computing errors. The balance between these two extremes is required. In practice, we often use a combination of the two methods – first choose a few of the median / maximum / minimum elements then use convolution with the kernel of their values. Anyway, some portions of data are always irrelevant or false, such data are to be filtered.

Window size selection. At small window sizes may not be enough data to make a decision, besides the role of inaccuracies and rough measurement errors increases. At large window sizes the greater impact may have pixels are not relevant to the object of interest, that can be solved with the help of masking or weighted summation; the greater impact may have imbalance. The practice of development of convolutional neural networks (as an example neural network VGG19 [6]) shows that the most successful decision is using of deep neural networks with great number of convolution layers (each subsequent layer takes as input the result of the previous layer) containing many small filter windows (3-by-3 pixels). Herewith max-pooling layer should be every 2–3 convolution layer. This result can be explained as follows. Let us say we use 3×3 pixels filter and each sell of filter binary mask may have the value of 0 or 1. Such filter has $2^9 = 512$ possible configurations. If we replace 3×3 pixels kernel on 5×5 pixels kernel there will be 2^9 possible configurations which is 2^{16} times larger. If we replace 5×5 pixels kernel on 7×7 pixels kernel difference will be even more essential. The intensity of image pixels within the kernel can take more than two values. Therefore it will require much more training data and sorting of a larger number of filter mask variants in the learning process. Besides, it increases the likelihood of filter adjustment to specific examples from training set. Natural disadvantage of using small kernels is reduction of receptive field (an image area that affects the value of the result). If the receptive field is not large enough the important parts of image and various global relationship between the local areas of the image just does not fall into the field of view of the algorithm. However, the more there are layers in neural network the larger the receptive field is. Thus, VGG neural networks have a great number of layers.

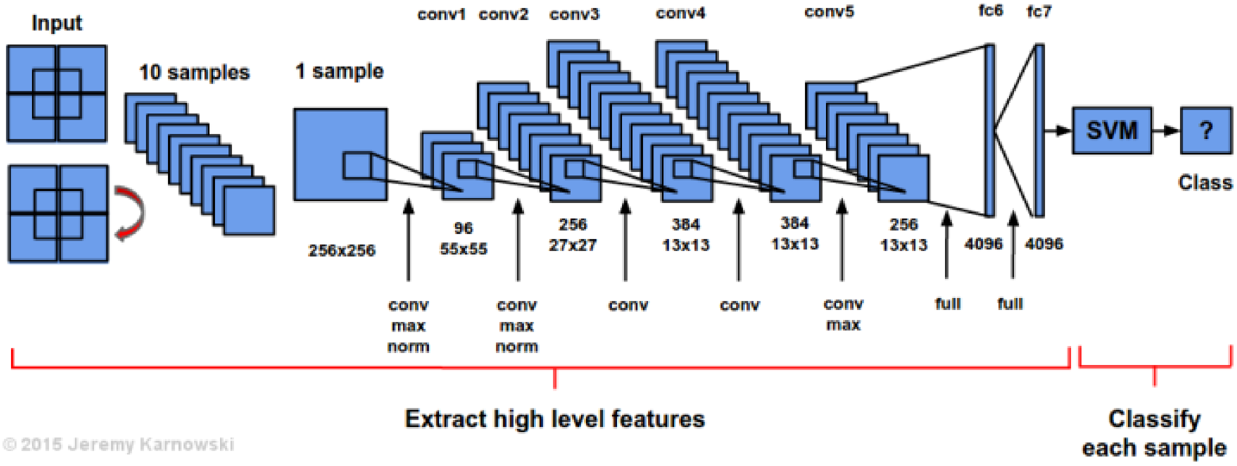

Fig. 1. Sample of computation scheme on an example of architecture of the convolutional neural network AlexNet [7]

Convolutional neural networks. Convolutional neural networks operation can be fully described on the basis of types of operations previously discussed, therefore we have all advantages and disadvantages of these operations. In convolutional layers, the operations of masking and summation are used. In max-pooling layers, selecting of one element is used. In fully connected layers, weighted summation and threshold condition are used (or nonlinear activation functions). In neurons of softmax layer, the maximum selecting operation is used.

3. The issues of recognition algorithms robustness

In the design of recognition systems very often an important role is played by the robustness of algorithm result and its stability over time. In the case of tracking systems development when the object of interest is stationary, from the recognition system is usually required that the result of algorithm operation should not significantly change (i.e. the output of the algorithm is not shaking). In the case of smooth movement of the object from algorithm is usually required the smoothness of the trajectory but not a hopping trajectory.

In development of smoothing algorithms (for robustness of result), the main role is played by properties of previously discussed computing operations. The most primitive method of increasing the robustness of result is using of the method of moving average. However, its application increases the delay time and system response to movement of the object, that not always acceptable. All the same, all outliers and inaccuracies in the measurements will affect the result. Now a sufficiently large number of more advanced data filtering techniques are developed [8]. More preferred is the following method of filtration: we leave the position of object from previous frame if position deviation at the new frame from the position at the previous frame does not exceed a certain threshold T that corresponds to the maximum tracking accuracy (to the value of recognizer noise).

4. Function randomization index

Randomization index $r(x)$ of function $f(x)$ for a given noise value n of input variable x is the sum of derivatives of this function on the interval $[x-n, x+n]$.

Randomization index of function $g(f(x))$ (when one function has as argument the result of another function) at point t is the sum of derivations of the values of final function $y=g(f(x))$ in the interval of variation of the input variable x with noise n $[t-n, t+n]$ from the value of final function $y=g(f(x))$ at point t.

Randomization index of computing scheme is the value of randomization index of the sequence of functions equivalent to a given computational scheme.

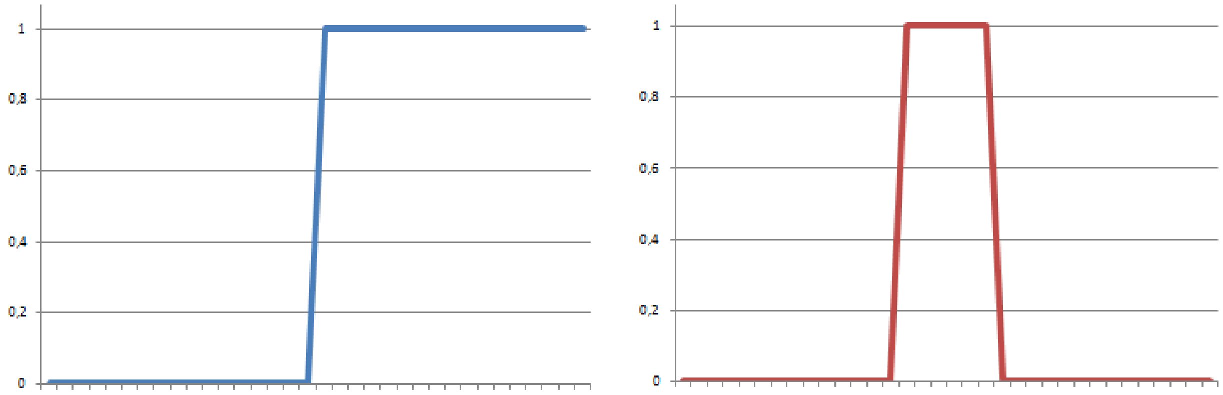

Fig. 2. Function and its randomization index

The physical meaning of randomization index is the variability of the result of function / computational scheme against small (within the noise / measurement error) changes in the input variable. The computational scheme with low randomization index will be more robust to small changes of input data and will give more stable results compared with computational scheme with high randomization index.

In particular, the threshold function (and similar functions, for example sigmoid function) has high randomization index near threshold. Thus, the using of threshold functions in computational scheme will lead to high randomization index at some points and as a result to instability of algorithm operation at the neighborhood of these points. Therefore, it is recommended to change the threshold functions for smoother analogs or averaging of several similar functions with different parameters (as in the case of using the decision forest instead of decision tree).

5. Practical issues

The features robustness is critically important in metric classification algorithms (*k-NN*, *SVM*, *LDA*, etc.). If features are not stable (in the presence of outliers and high dispersion), the objects will be dispersed at the features space and it will be difficult to construct the separating surface between classes. With the increase in the number of features the object location chaos will only increase (sometimes it is called the “curse of dimension”). Therefore, in particular, the *SVM* algorithm is seldom used by itself but more often is used over convolutional neural networks (the features are the values of network outputs, such features can be considered as rather robust). The best solution for the synthetic data with well-separable classes will likely be a metric classifier.

To the issues of learning algorithm robustness we can include the classic example – the fitting of polynomials of varying degrees to a set of points – with the help of which the phenomenon of overtraining is usually explained. High degree polynomial fitting has higher randomization index than low degree polynomial fitting. The selection of higher-degree polynomial with the same degree of accuracy requires much more computing. Let's say we choose a first degree polynomial with 2 coefficients, for each coefficient we iterate 10 values, there will be 100 variants at the whole. Let’s then say we want to choose a second degree polynomial with 3 coefficients. In order to provide the same guaranteed accuracy of the device we are to go through 1000 variants, otherwise because of the loss of precision the approximation error on the test data may increase. As we see, the discussed example does not refer to overtraining because of too accurate learning of training data but does refer to the use of computational operations in the learning process.

Unrobust behavior is demonstrated by the straight-line recovery of two closely spaced points. Small changes in positions of points relative to each other will cause significant change in the angle of rotation of straight line and if it is necessary to recover long line its ends will drastically change their position. In the same time the recovery of straight line parameters on a small number of points are widely used in computer vision tasks, for example in the case of recovery of object position on its singular points by the *RANSAC* algorithm [9] or when calculating directions of line segments on the basis of gradients. Therefore, the phenomenon described above should be taken into account in the development of recognition systems.

Conclusions

The article provides an overview of the properties of computing operations, which are most common in computer vision tasks. The analysis of the robustness of these operations in relation to the most common types of input data mismatches is done. The practical importance of the analysis of these properties is demonstrated. The article deals with the practical issues of development of machine learning systems and recognition systems from point of view of computational processes and their features. An algorithm for smoothing the recognition results to protect against jitters is suggested.

References

1. Leibe B., Leonardis A., Schiele B. An Implicit Shape Model for Combined Object Categorization and segmentation. Springer Berlin Heidelberg, 2006, pp. 508–524.

2. Gonsales R. *Tsifrovaya obrabotka izobrazheniy* [Digital Image Processing]. Moscow, Tehnosfera, 2005. 1072 p.

3. Nagi J., Ducatelle F., DiCaro G.A., Ciresan D., Meier U., Giusti A., Nagi F., Schmidhuber J., Gambardella L.M. Max-Pooling Convolutional Neural Networks for Vision-Based Hand Gesture Recognition. *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, IEEE, 2011, pp. 342–347.

4. Magee J.F. Decision Trees for Decision Making. *Harvard Business Review*, 1964, vol 42, no. 4, pp. 126–138.

5. Krizhevsky A., Sutskever I., Hinton G.E. Imagenet Classification with Deep Convolutional Neural Networks. *NIPS 2012: Neural Information Processing Systems*, Lake Tahoe, Nevada, 2012, pp. 1097–1105.

6. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv technical report:1409.1556*, 2014.

7. Karnowski E. Alexnet visualization. July 15, 2015. Available at: https://jeremykarnowski.wordpress.com/2015/07/15/alexnet-visualization/.

8. Azimi M. Skeletal Joint Smoothing White Paper. Available at: https://msdn.microsoft.com/en-us/library/jj131429.aspx.

9. Fischler M.A., Bolles R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 1981, vol. 24, no. 6, pp. 381–395.

Received 6 November 2016

УДК 004.855.5 DOI: 10.14529/ctcr170114

СВОЙСТВА ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ В ЗАДАЧАХ АНАЛИЗА ИЗОБРАЖЕНИЙ И МАШИННОГО ОБУЧЕНИЯ

И.В. Парасич, А.В. Парасич

Южно-Уральский государственный университет, г. Челябинск

Процесс решения любой задачи компьютерного зрения или машинного обучения можно представить в виде некоторой последовательности вычислительных операций над набором входных данных. Особенностью задач интеллектуального анализа данных является существенная неоднородность входных данных – могут присутствовать выбросы, неточность измерений, мультимодальность. Разные типы вычислительных операций по-разному реагируют на представленные типы рассогласований. При этом от свойств базовых операций и их устойчивости к рассогласованиям в данных во многом зависит качество решения задачи. В статье рассматриваются основные типы вычислительных операций, применяемых в алгоритмах компьютерного зрения и машинного обучения, проводится анализ их устойчивости к различным типам рассогласований в данных. Рассмотренная информация будет полезна при проектировании дескрипторов визуальных объектов, алгоритмов распознавания и трекинга объектов. Особую ценность представляет применение рассмотренной информации к проектированию и анализу глубоких сверточных нейронных сетей.

Ключевые слова: компьютерное зрение, машинное обучение, сверточные нейронные сети, трекинг, фильтрация.

Литература

1. Leibe, B. An implicit shape model for combined object categorization and segmentation / B. Leibe, A. Leonardis, B. Schiele. – Springer Berlin Heidelberg, 2006. – P. 508–524.

2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с.

3. Nagi, J. Max-pooling convolutional neural networks for vision-based hand gesture recognition / J. Nagi, F. Ducatelle, G.A. DiCaro et al. // IEEE International Conference on Signal and Image Processing Applications (ICSIPA) – IEEE, 2011. – P. 342–347.

4. Magee, J.F. Decision trees for decision making / J.F. Magee // Harvard Business Review. – 1964. – Vol. 42, no. 4. – pp. 126–138.

5. Krizhevsky, A. Imagenet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G.E. Hinton // NIPS 2012: Neural Information Processing Systems. – Lake Tahoe, Nevada, 2012. – P. 1097–1105.

6. Simonyan, K. Very deep convolutional networks for large-scale image recognition / K. Simonyan, A. Zisserman // arXiv technical report:1409.1556. – 2014.

7. Karnowski, E. Alexnet visualization / E. Karnowski. – July 15, 2015. – https://jeremykarnowski.wordpress.com/2015/07/15/alexnet-visualization/

8. Azimi, M. Skeletal joint smoothing white paper / M. Azimi. – https://msdn.microsoft.com/en-us/library/jj131429.aspx

9. Fischler, M.A. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography / M.A. Fischler, R.C. Bolles // Communications of the ACM. – 1981. – Vol. 24, no. 6. – P. 381–395.

Парасич Ирина Васильевна, канд. техн. наук, доцент кафедры математического и компьютерного моделирования, Южно-Уральский государственный университет, г. Челябинск; parasichiv@ mail.ru.

Парасич Андрей Викторович, аспирант кафедры электронных вычислительных машин, Южно-Уральский государственный университет, г. Челябинск; parasich_av@yandex.ru.

Поступила в редакцию 6 ноября 2016 г.

ОБРАЗЕЦ ЦИТИРОВАНИЯ

Parasich, I.V. The Properties of Computing Processes in Image Analysis and Machine Learning Tasks / I.V. Parasich, A.V. Parasich // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». – 2017. – Т. 17, № 1. – С. 126–133. DOI: 10.14529/ctcr170114

FOR CITATION

Parasich I.V., Parasich A.V. The Properties of Computing Processes in Image Analysis and Machine Learning Tasks. *Bulletin of the South Ural State University. Ser. Computer Technologies, Automatic Control, Radio Electronics*, 2017, vol. 17, no. 1, pp. 126–133. DOI: 10.14529/ctcr170114